mainfont: “DejaVu Sans”
monofont: “DejaVu Sans Mono”

Software Architecture Document (SAD)¶

Weather Ready Pacific - Integrated Forecasting Platform Landscape¶

Project: Integrated Forecasting Platform
Date: 2026-06-23
Version: 1.1
Status: ACTIVE (Living Document)
Scope: Three integrated software systems (IFP, Regional WIS2.0 Hub Node, PDTS)

Recent Updates (v1.1):

- Clarified MQTT authentication model (no ACLs; everybody:everybody placeholder credentials)
- Added Bearer Token Management section (2.2.5)
- Clarified IFP subscribes to Global WIS2 Broker (not individual PICT nodes)
- Updated PDTS private broker to use username/password authentication

Executive Summary¶

The **Integrated Forecasting Platform (IFP)** is the primary software system for the Weather Ready Pacific initiative, tasked with generating authoritative meteorological forecasts, warnings, and decision-support products for the Pacific region.

To deliver comprehensive forecasting capability, IFP requires **meteorological data from multiple sources:**

- **PICT observations** (local surface, radar, upper-air, tide gauges) via Regional WIS2 Hub Node or country-owned nodes
- **Global atmospheric predictions** (NWP models, satellite imagery) via Pacific Data Tailoring Service
- **Global observations** (background conditions) via Global WIS2 Network

Three integrated software systems work together to deliver this data to IFP:

1. **IFP (PRIMARY SYSTEM)** - Forecast generation, visualisation, warnings
2. **Regional WIS2.0 Hub Node (SUPPORTING)** - PICT data publishing, WIS2 compliance
3. **Pacific Data Tailoring Service (SUPPORTING)** - Global data retrieval, geospatial optimisation, private data pipeline

This document describes the complete integrated architecture, including:

- System landscape and interdependencies
- Data integration patterns (four converging data paths)
- Container-level architecture for each system
- Architectural decisions documented in ADRs
- Operational glossary and terminology

Table of Contents¶

Table of Contents *generated with DocToc*

Section 1: System Landscape & Mission¶

1.1 Weather Ready Pacific Initiative¶

The **Weather Ready Pacific (WRP)** initiative is a regional effort to improve meteorological capabilities for Pacific Island Countries and Territories (PICTs) through enhanced forecasting, warning systems, and climate resilience.

Core Challenge:

- PICTs have varying levels of meteorological infrastructure and technical capacity
- No single country has sufficient resources to maintain independent NWP capability
- Global forecasting models require geospatial optimisation for Pacific region
- PICT data sovereignty must be protected while enabling regional collaboration

Solution Architecture: Pacific Region Integrated Forecasting Platform and Data Flow Architecture

- **Regional data broker** (WIS2 Node) consolidates PICT observations for Pacific region
- **Integrated forecasting platform** (IFP) synthesizes all data sources for forecast generation

This architecture enables:

- **Global data optimizer** (PDTS) provides tailored NWP/satellite

✅ **Data sovereignty** - PICT data remains under country/PICT control
✅ **Operational efficiency** - Shared infrastructure reduces per-country burden
✅ **Forecast quality** - IFP access to comprehensive meteorological data
✅ **WIS2 compliance** - Regional participation in global standards

---

1.2 IFP Role¶

The **Integrated Forecasting Platform (IFP)** is the central operational system that enables the WRP forecasting initiative's capability.

IFP Responsibilities:

- ✅ Ingest meteorological observations and global data
- ✅ Transform data into standardized internal formats
- ✅ Store data in a unified Data Lake
- ✅ Visualize data for forecasters
- ✅ Generate forecasts (public weather, marine, aviation, severe weather, geological hazards, hydrological and tropical cyclone forecasting services) along PICT based workflows
- ✅ Generate products (e.g. CAP alerts, warnings, advisories and other formats including ICAO/WMO compliant outputs.)
- ✅ Disseminate products to operational consumers along PICT based delivery channels

What IFP Does NOT Do:

- ❌ Raw data ingestion from PICT observation equipment (handled by the PICT data services such as CliDE or equivalent and then WIS2 Node or PDTS)
- ❌ Stage 1 quality control on raw observations (handled by PICT data services such as CliDE or equivalent instances)
- ❌ WIS2 publication (handled by PICT WIS2 Node or Regional WIS2 Hub Node)
- ❌ Global Data set optimisations (handled by PDTS)

This separation of concerns ensures:

- **Scalability** - WIS2 Node handles ingestion scale independently of IFP capacity
- **Reliability** - Data ingestion failures don't halt IFP forecast generation
- **Maintainability** - Each system has clear ownership of specific responsibilities
- **Operational focus** - IFP team focuses on forecasting; WIS2 team focuses on data standards

---

1.3 Two Primary User Personas¶

The IFP serves **two distinct forecasting personas** with different operational constraints and geographic scales. These personas are derived from analysis of 15+ requirements-based roles identified across use cases, surveys, and stakeholder documents. Both personas consolidate multiple specialized responsibilities into unified roles aligned with real-world operational workforce realities.

RSMC Forecaster (Fiji) - Regional Scale¶

Also Known As: Regional forecasting team at RSMC Fiji
Consolidates: 6+ requirements roles including Forecast Duty Officer (regional), Aviation/Marine Forecasters (regional), TC Advisory Author, Solution Architect, Director/Ops Manager

Geographic Context: Fiji (on-premises)
Workforce: Part of larger RSMC team
Client: Visual Weather Server (Fat Client - high-performance desktop)
Network: On-premises Fiji data center LAN (high bandwidth, low latency)

Primary Responsibilities:

- Regional weather forecast analysis and generation (2-4x daily)
- Tropical Cyclone advisory creation, update, and approval
- Common Alerting Protocol (CAP) alert generation (integrated workflow)
- Regional product coordination and quality assurance
- Reference authority for National NMHS Forecasters
- Multi-specialisation: public weather, marine, aviation, severe weather, geological hazards, hydrological and tropical cyclone forecasting services.

Operational Constraints & Technical Needs:

- **High-Performance Requirement:** Intensive NWP model processing, multiple simultaneous forecasts, interactive data transformation
- **Comprehensive Data Access:** Must access all data classifications to ensure forecast quality

- **Regional Authority:** Single point of accountability for regional products
- **Technical Environment:** On-premises infrastructure, high-speed LAN, computational resources available

Reference: See IFP_PERSONAS Section 1 for detailed workflows and tasks.

National NMHS Forecaster - PICT Scale¶

Also Known As: Country-level meteorologist in PICTs
Consolidates: 6+ requirements roles including Forecast Duty Officer (country), Aviation/Marine/Hydro Forecasters (country), Observing Equipment Operator, Communications Officer

Geographic Context: Distributed across 15 Pacific Island Countries
Workforce: 1-4 dedicated meteorologists per country (survey evidence from 8 PICTs)
Client: Visual Weather Web Client (Single Page web Application 'SPA' - lightweight browser-based)
Network: Internet access

Primary Responsibilities:

- Local forecast analysis and generation (1-2x daily)
- Integration of regional RSMC forecasts into local context
- Consumption of CAP alerts and TC advisories from RSMC
- Local data quality feedback and monitoring
- Multi-specialisation: general forecasting, aviation, marine, hydrological analysis (limited staffing)
- Potential future responsibilities: local warning generation (Phase 2+)

Operational Constraints & Technical Needs:

- **Limited Bandwidth:** Internet-dependent access (6-110 Mbps across PICTs, with median ~40-50 Mbps), not LAN speeds; many countries constrained at 6-50 Mbps; cannot transfer large data volumes efficiently
- **Limited Hardware:** Older computers, limited RAM/GPU (survey: outdated hardware is #1 constraint cited by PICTs)
- **Limited Staffing:** Typically 1-4 meteorologists per country must perform multiple specialisations; cannot afford specialist roles

- **Data Sovereignty:** PICT retain ownership and determine terms for usage; private observations restricted to country level workflow and usage
- **Simplified Workflows:** Need templated, automated workflows rather than intensive manual analysis

Architectural Impact: These constraints drive the **Open Weather Server (lightweight web SPA) backend design**, requiring OGC API standards (web protocols), server-side rendering (reduce data transfer), async loading, and simplified templated workflows for ease of use and reduced training burden.

> **Bandwidth Evidence:** IFP Survey (11 PICTs, 2025-10-30 to 2025-11-05) reports: Niue 6-8 Mbps, PNG/Solomon Islands ~50 Mbps, Tuvalu 20-60 Mbps (Starlink), Palau/Nauru 27-44 Mbps, Tonga 80 Mbps (fiber), Vanuatu/Cook Islands 93-110+ Mbps. Lower-bandwidth countries (≤50 Mbps) represent ~60% of operational PICTs; this constraint remains critical for regional architecture.

Survey Evidence (IFP Survey of 8 PICTs):

Country	Forecasters	Current System	Key Challenge
Vanuatu	4-10	Manual + visualisation tools	Limited specialized expertise
Solomon Islands	3-4	Manual + internet visualisation	Manual data integration
PNG	3+	Manual + word/html/excel	No structured platform
Niue	1	Manual + email	Single operator (no redundancy)
Kiribati	2-3	Legacy systems	Outdated infrastructure
Tonga	2-3	Varies	Limited capacity
Cook Islands	2-3	Varies	Connectivity constraints
Palau	1-2	Manual + visualisation	Minimal staffing
AVERAGE	**1-4**	**Limited/Manual**	**Multi-disciplinary required**

Key Implications:

- Most PICT services cannot afford separate aviation/marine/hydro forecasters
- Forecasters are **multi-disciplinary generalists** (not specialists)
- Lightweight web interface is **critical** (not optional)
- Simplified workflows and templates are **essential** for adoption
- Current systems are mostly manual; IFP will be their **first integrated platform**

Reference: See IFP_PERSONAS Section 2 for detailed workflows and tasks.

Architectural Implications: Why Dual Backends?¶

The two personas have **incompatible operational constraints** that cannot be served by a single backend:

Aspect	RSMC Forecaster	National NMHS Forecaster	Architectural Response
Access Pattern	On-premises LAN	Remote internet	Visual Weather Server vs. Open Weather Server
Performance Needs	High-intensive analysis	Lightweight/templated	Fat client vs. Web SPA
Bandwidth Constraint	None	6-110 Mbps (median 40-50)	Different optimisation strategies
Data Access	All classifications	Country-specific only	Super-Consumer vs. Data Sovereignty TODO
Specialisation	Can be regional specialists	Must be multi-disciplinary	Different workflow designs

Both backends access data using the same architectural components:

- The **a unified Data Lake** (S3-compatible object storage using RustFS)
- The **same forecast models** and business logic (Python Engine, shared algorithms)
- The **same dissemination layer architecture** (products published once, consumed by both)

The **dual backends provide different access patterns** optimized for each persona’s operational reality, not duplicated forecasting logic.

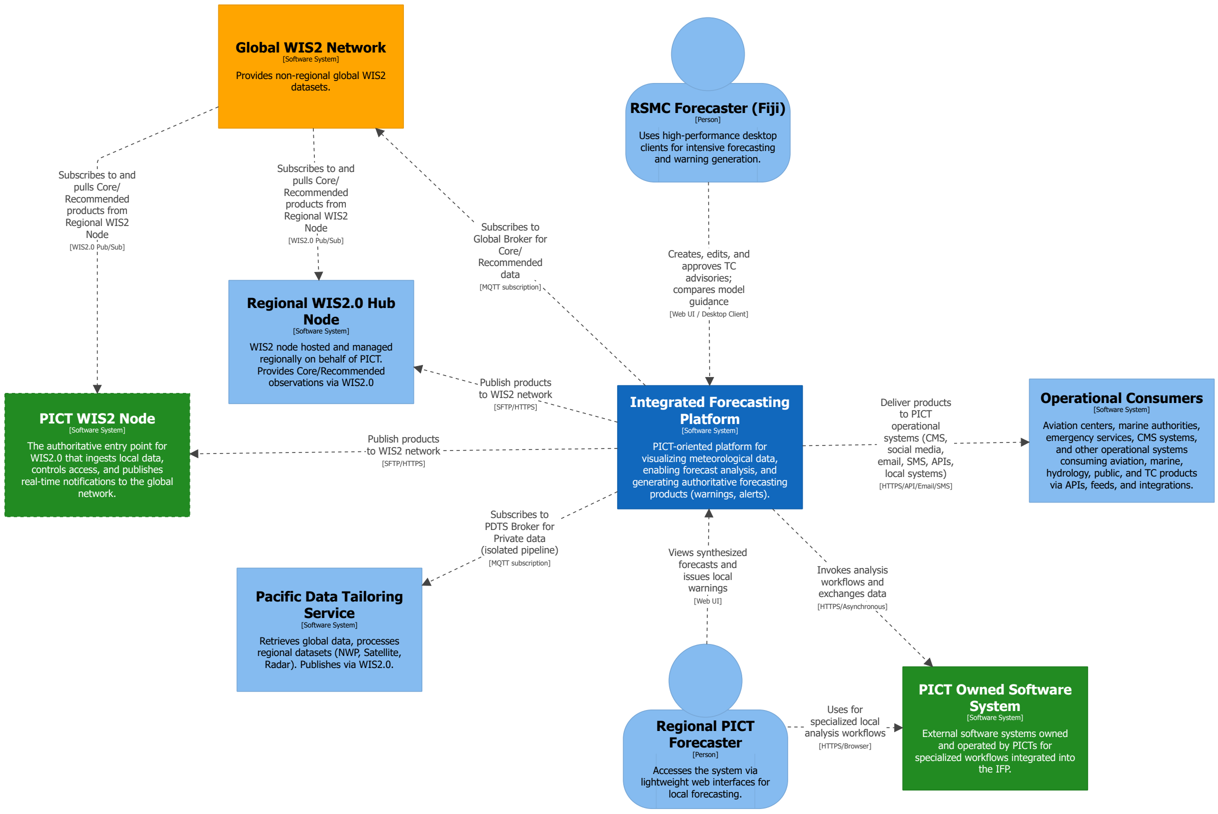
Related Architectural Decision: See ADR-0012 (Dual Backend Architecture) for complete rationale and consequences.

1.4 Three Integrated Systems Required¶

To deliver comprehensive meteorological data to IFP for forecast generation, **three interdependent software systems** are required:

System 1: IFP (PRIMARY)¶

Aspect	Description
Role	Forecast generation, visualisation, warnings
Key Responsibility	Consume all data sources; visualize for forecasters; generate forecasts
Deployment	Central regional platform (AWS or on-premises)
Data Consumption	Global WIS2 Network + PDTS private pipeline + direct S3 access
Data Output	Forecasts, warnings, products (disseminated via HTTPS/API)
Personas Served	RSMC Forecaster, National NMHS Forecaster
Geographic Scope	Regional (Pacific-wide)



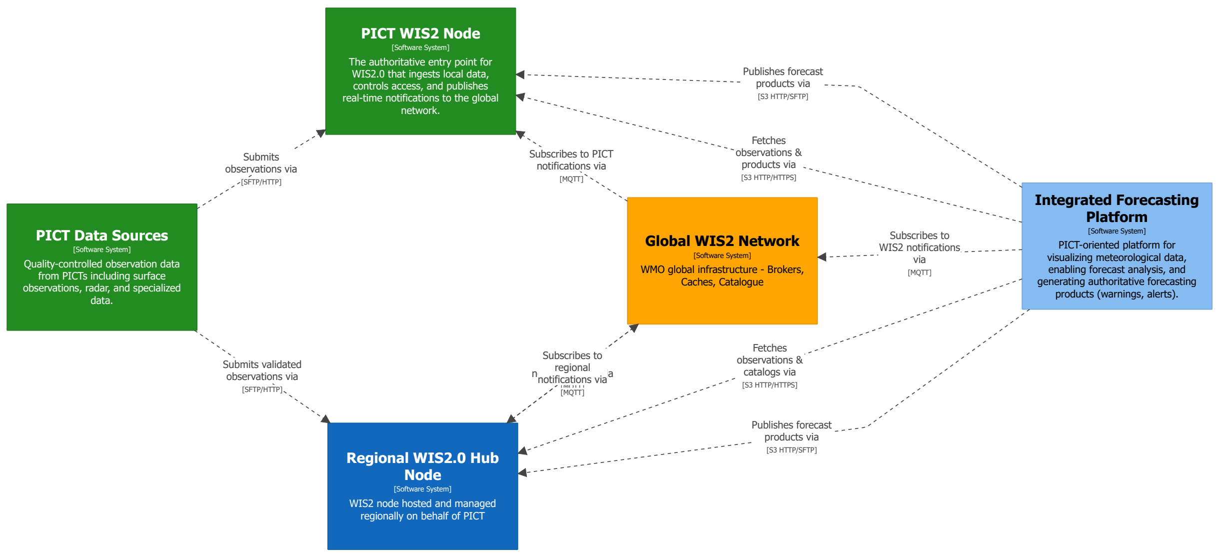
System Context View: Integrated Forecasting Platform
Thursday, June 11, 2026 at 4:34 PM New Zealand Standard Time

System 2: Regional WIS2.0 Hub Node (SUPPORTING)¶

Aspect	Description
Role	Where the PICT is not able to implement their own WIS2 Node, this provides a mechanism for the PICT to get their data into the IFP, via Global WIS2 network.
Key Responsibility	Ingest PICT observations; apply QC; publish to Global WIS2 Network
Deployment Options	Option 1: Single shared hub (Fiji); Option 2: Per-PICT nodes (regional)
Data Input	PICT observations (surface, upper-air, radar, tide gauges) via SFTP/S3
Data Output	PICT Core/Recommended data published to Global WIS2 Network
Standards	WIS2.0 compliance; WMO data classifications
Technology	wis2box reference implementation (MinIO); MQTT broker
Governance	SPREP (Option 1) or PICT-specific (Option 2)

Data Sovereignty:

- Enforces path isolation in S3-compatible storage (centre-id separation)
- Access control at file download layer (HTTPS bearer tokens for Recommended data)
- Private observations NOT published to WIS2Node (kept sovereign via PDTS private pipeline)

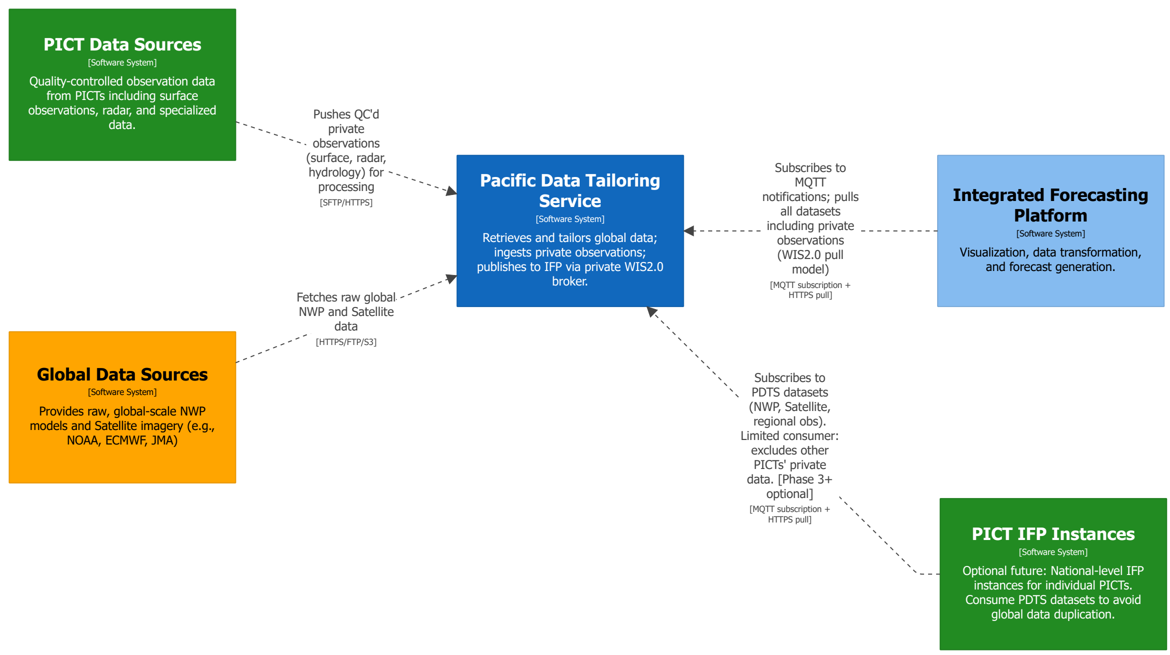


Regional WIS2.0 Hub Node - System Context
Wednesday, May 27, 2026 at 2:05 PM New Zealand Standard Time

System 3: PDTS (SUPPORTING)¶

Aspect	Description
Role	Global data retrieval, geospatial optimisation, private data handling
Key Responsibility	Fetch global NWP/satellite; optimize for Pacific region; ingest PICT private obs; publish to IFP-only channel
Deployment	AWS ECS + Fargate (serverless, auto-scaling)
Data Input	Global NWP models (NOAA GFS, ECMWF, JMA); Satellite (Himawari-8); PICT private obs (radar, hydrology)
Data Output	Optimized global data + private obs via Private PDTS wis2box (IFP-only)
Geospatial Optimisation	Crops global grids to Pacific region (~95% volume reduction)
Data Isolation	All PDTS outputs marked Private; never published to Global WIS2

Aspect	Description
Operations	SPREP / Regional team



System Context View: Pacific Data Tailoring Service
Wednesday, May 20, 2026 at 3:04 PM New Zealand Standard Time

1.5 Why All Three Systems?¶

The Data Problem¶

IFP needs comprehensive meteorological data for high-quality forecasts:

Data Category	Sources	Latency	Sovereignty
PICT Observations	Surface stations, radiosondes, radar, tide gauges	hourly	PICT-owned
Global NWP Models	NOAA GFS, ECMWF, JMA (global grids)	6-12h	No sovereignty concern
Regional/Local Models	ESNZ wave models, WRP-funded regional NWP, PICT-developed hazard models	Varies (6-24h)	Varies by provider

Data Category	Sources	Latency	Sovereignty
Satellite Imagery	Himawari-8, NOAA satellites	Varies (10 min - hourly)*	No sovereignty concern
Global Observations	WMO global network	Real-time	Global sharing

⚠️ **DATA VOLUMES TO BE DETERMINED:** These values need to be sourced to inform infrastructure procurement decisions and tailoring pipelines:

- **PICT Observations:** Need estimates and actual history from RSMC Fiji and other PICT observation systems
- **Global NWP Models:** Confirm parameter subset and resolution from NOAA/ECMWF (full models: ~100 GB/week; subset: 30-50 GB/week)
- **Regional/Local Models:** Need provider confirmation on model availability, format, and update frequency
- **Satellite Imagery:** Decide scope (Pacific subset and frequency)
- **Global Observations:** Validate WMO GTS baseline (~16 GB/month typical; may vary by feed selection)

📋 **Design Proposals for Regional/Local Model Integration:**

The integration pathway for regional and local models (e.g., ESNZ wave models, WRP-funded regional NWP) is documented in separate design proposals:

- **Design Proposal: Regional and Local Model Integration** — PDTS as unified ingestion point for regional model outputs
- **Design Proposal: Regional/Local Model Scenarios** — Decision matrix for model execution location, distribution mechanism, and data classification

These proposals address: how regional models are ingested (PDTS fetch vs. PICT push), data classification (RECOMMENDED vs. RECOMMENDED*), and operational governance.

The Three-System Solution¶

WIS2 Node handles PICT data delivery:

- Consolidates all PICT observations into a WMO standardized system
- Publishes to Global WIS2 Network (standard WIS2 consumers get same data)

- Enforces data sovereignty (CORE and RECOMMENDED) data are clearly catagorised and attributed to issuing PICT
- IFP subscribes like any other WIS2 consumer

PDTS handles global data optimisation:

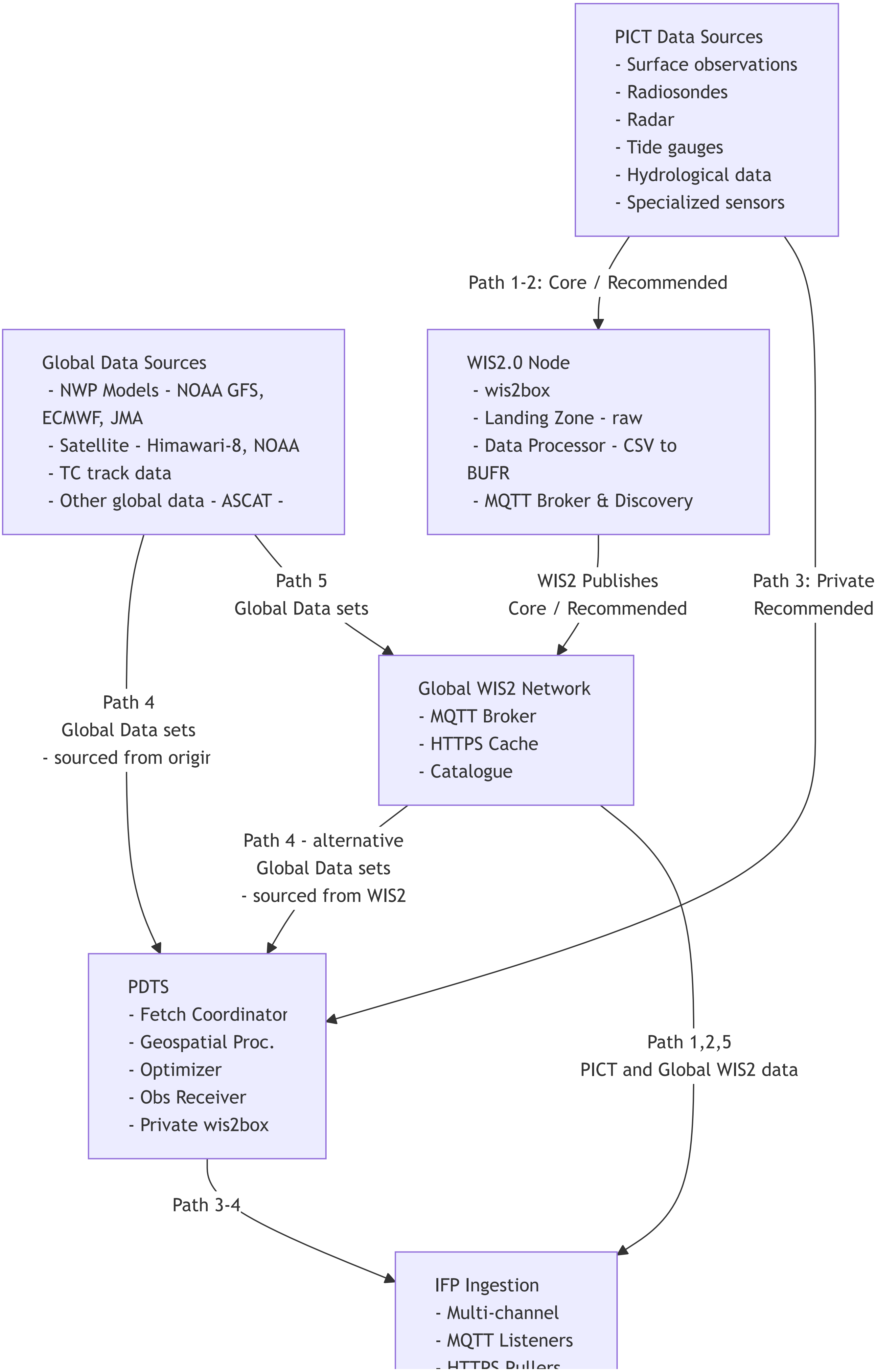
- Fetches massive global datasets from external sources, including Global WIS2
- Optimises data sets for the IFP through geospatial cropping (e.g to Pacific-only regions to reduce data size)
- A private pipeline to the IFP for PICT Private Data Sets (e.g. Commercial Obs)
- Publishes optimized data to private IFP channel (never globally)

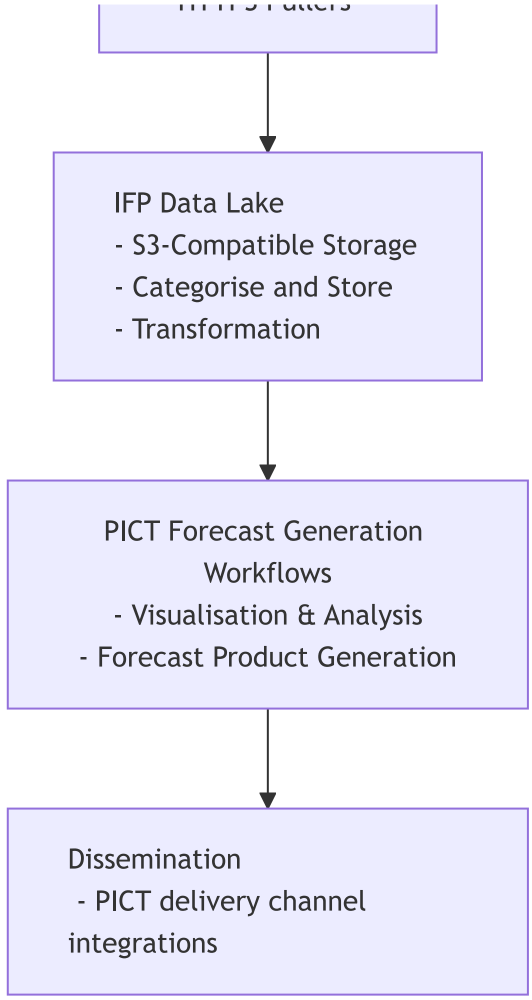
IFP focuses on forecasting:

- Consumes standardized data from two channels (Global WIS2 + PDTS private)
- Transforms to internal formats if required
- Provides PICT specific workflows for forecasting workflows and forecast product generation

---

1.6 System Interdependencies¶


PICT Data Sources
- Surface observations
- Radiosondes
- Radar
- Tide gauges
- Hydrological data
- Specialized sensors
Path 1-2: Core / Recommended
Global Data Sources
- NWP Models - NOAA GFS, ECMWF, JMA
- Satellite - Himawari-8, NOAA
- TC track data
- Other global data - ASCAT -
WIS2.0 Node
- wis2box
- Landing Zone - raw
- Data Processor - CSV to BUFR
- MQTT Broker & Discovery
Path 5
Global Data sets
WIS2 Publishes
Core / Recommended
Path 3: Private
Recommended
Global WIS2 Network
- MQTT Broker
- HTTPS Cache
- Catalogue
Path 4
Global Data sets
- sourced from origir
Path 4 - alternative
Global Data sets
- sourced from WIS2
PDTS
- Fetch Coordinator
- Geospatial Proc.
- Optimizer
- Obs Receiver
- Private wis2box
Path 1,2,5
PICT and Global WIS2 data
Path 3-4
IFP Ingestion
- Multi-channel
- MQTT Listeners



Path Definitions:

Path 1-2: PICT Public Data sets (e.g. Observations)

- PICT Data source → WIS2 Node(PICT or Regional Hub) → Global WIS2 → IFP (Core/Recommended; sovereignty preserved)

Path 3: Global Data sets requiring tailoring for region (e.g. NWP, Satellite)

- Global Data source (Origin or Global WIS2 network) → PDTS (Ingestion & Optimisation) → Private wis2box → IFP (geospatially optimized; IFP-only access)

Path 4: PICT Private Data Sets (e.g. Commercial Obs)

- PICT data source → PDTS (Ingestion & Optimisation) → Private wis2box → IFP (geospatially optimized; IFP-only access)

Path 5: Global Data sets that do not require tailoring for region(e.g. Global Obs)

- Global Data source → Global WIS2 → IFP

Section 2: System Context & Data Integration¶

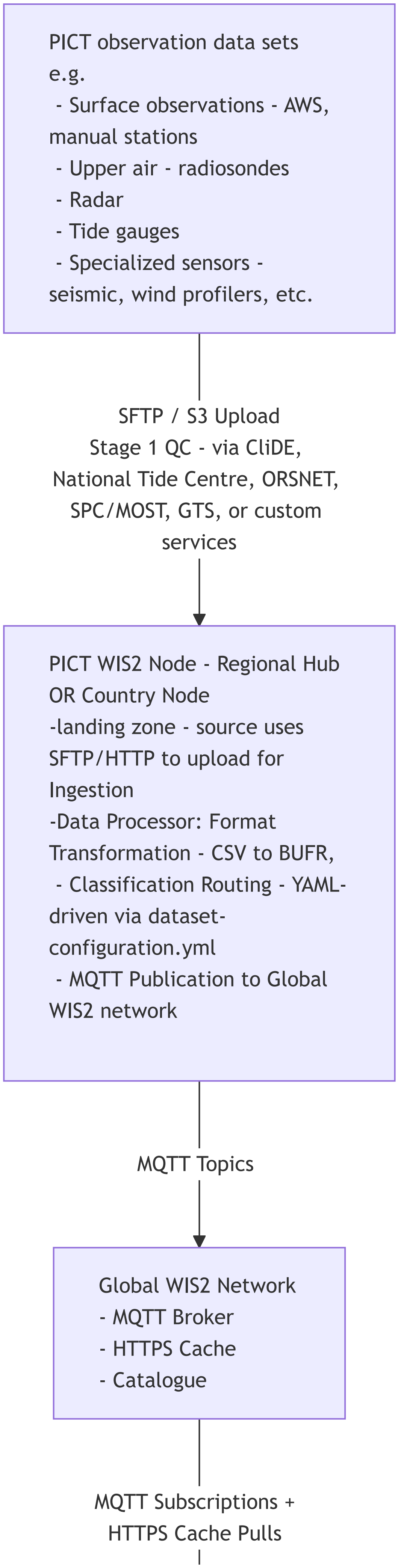
2.1 Four Converging Data Paths¶

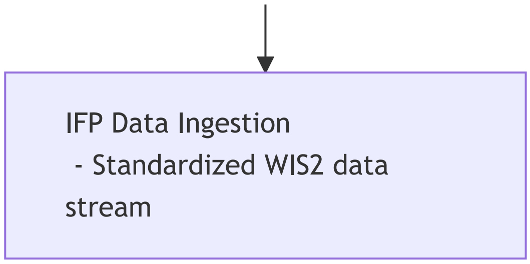
IFP synthesizes meteorological data via **four distinct data paths**, each serving a specific purpose:

Path 1-2: PICT Public Data sets (e.g. Observations)¶

PICT Public Data sets (e.g. Observations) are routed to the IFP via the Global WIS2 network either by:

- **Path 1:** PICT owned WIS2 Node
- **Path 2:** Shared Regional WIS2 node


PICT observation data sets e.g.
- Surface observations - AWS, manual stations
- Upper air - radiosondes
- Radar
- Tide gauges
- Specialized sensors - seismic, wind profilers, etc.
SFTP / S3 Upload
Stage 1 QC - via CliDE, National Tide Centre, ORSNET, SPC/MOST, GTS, or custom services
PICT WIS2 Node - Regional Hub OR Country Node
-landing zone - source uses SFTP/HTTP to upload for Ingestion
-Data Processor: Format Transformation - CSV to BUFR,
- Classification Routing - YAML-driven via dataset-configuration.yml
- MQTT Publication to Global WIS2 network
MQTT Topics
Global WIS2 Network
- MQTT Broker
- HTTPS Cache
- Catalogue
MQTT Subscriptions + HTTPS Cache Pulls



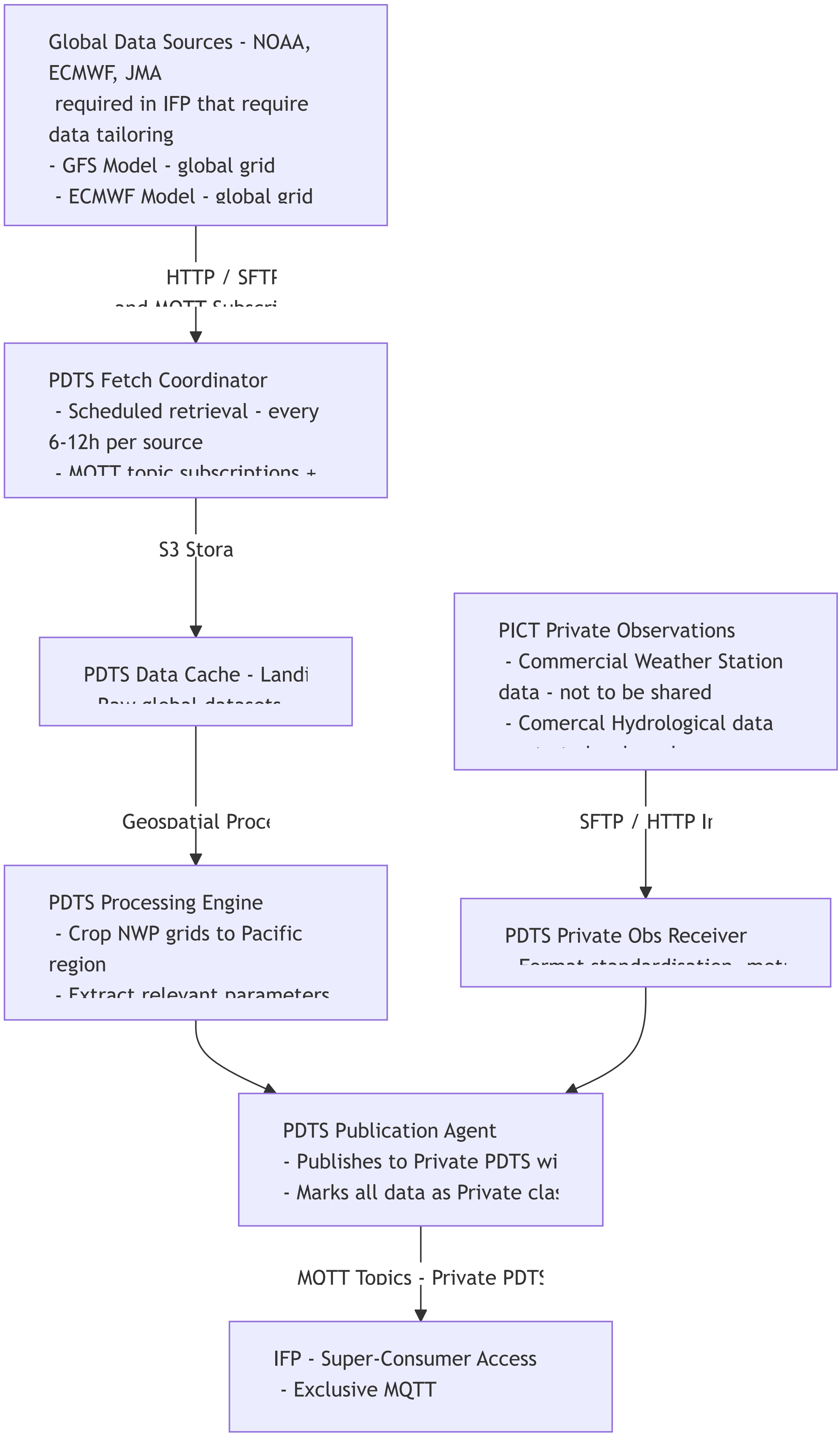
Key Characteristics:

- **Standard mechanism:** PICT Pipeline to the IFP using standard WIS2 path to Global WIS2
- **Data classification:** Core and Recommended
- **Access control:** IFP is standard WIS2 subscriber (like any global consumer)
- **Why:** WIS2 compliance; data sovereignty; operational decoupling

ADR Reference: `IFP/decisions/0002-IFP-wis2-ingestion-mechanism.md`

Path 3 and 4: Global Data sets + PICT Private Data Sets(e.g. Commercial Obs)¶

- **Path 3:** Global Data sets requiring tailoring for region (e.g. NWP, Satellite)
- **Path 4:** PICT Private Data Sets (e.g. Commercial Obs) via private data pipeline



Key Characteristics:

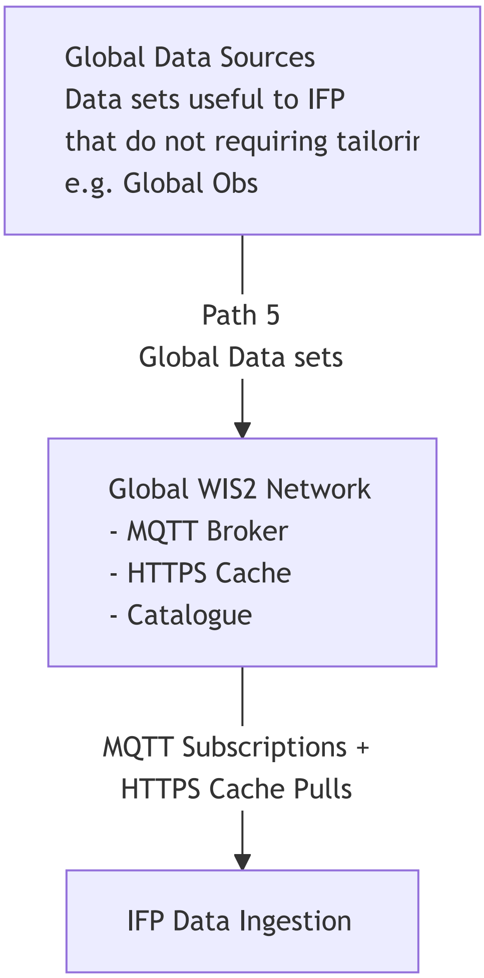
- **Optimisation:** volume reduction via geospatial cropping
- **Sovereignty protection:** PICT private data sets kept private (IFP-only access)
- **Isolation:** Private wis2box is internal to PDTS-IFP relationship
- **Standardisation:** IFP ingestion from PDTS the same as from Global WIS2 (MQTT + HTTPS) same as from Global WIS2 (MQTT + HTTPS)

ADR References:

- `IFP/decisions/0001-IFP-ifp-super-consumer-authorisation.md` (why IFP can access private data)
- `IFP/decisions/0005-IFP-dual-data-ingestion-sources-wis2-and-pdts.md` (why both paths needed)

Path 4: Global Data sets that do not require tailoring for region¶

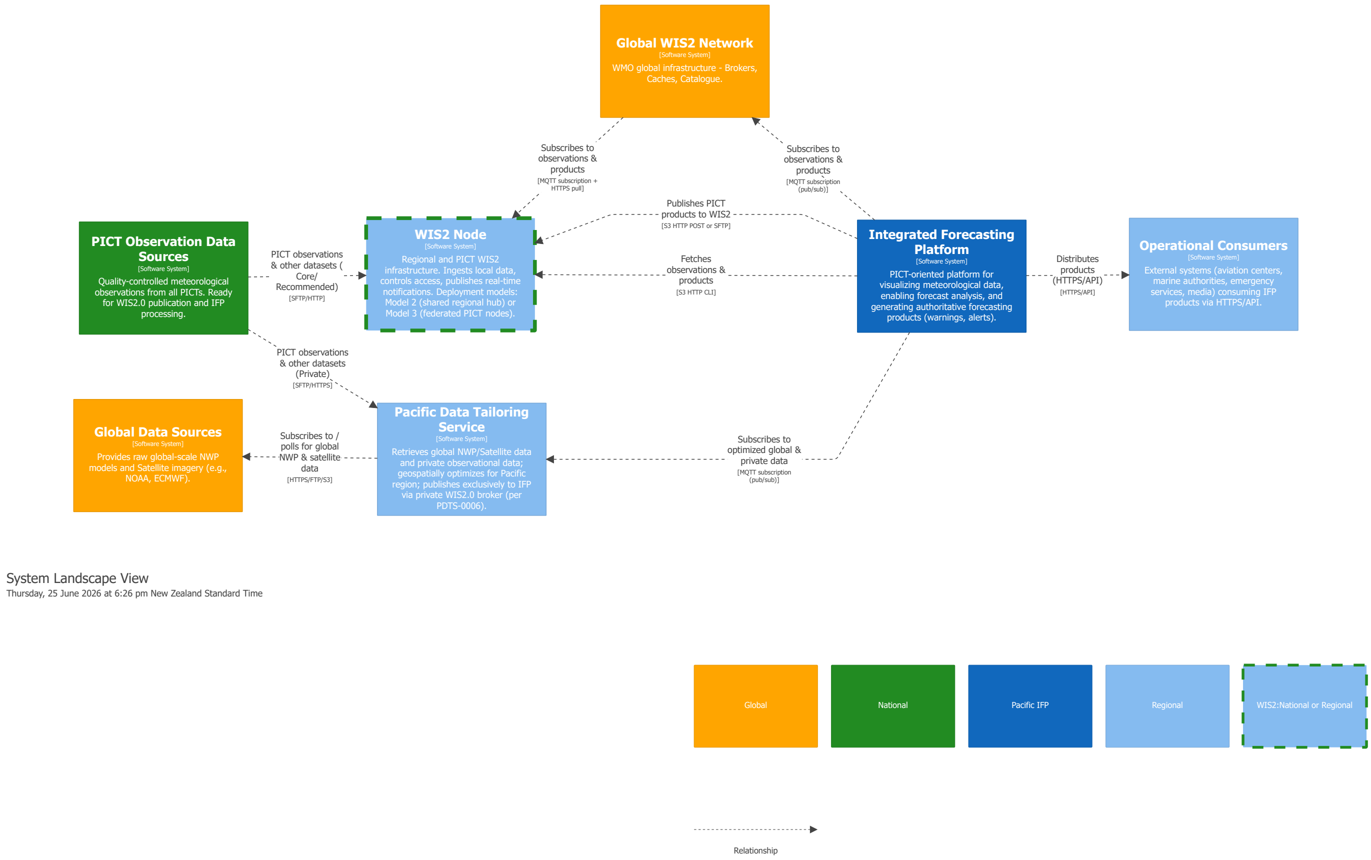
Data sets that IFP can ingest directly from WIS2 (e.g. Global Obs)



Key Characteristics:

- **Passive consumption:** IFP receives data published by others
- **Background context:** Provides hemispheric/global context
- **No PDTS involvement:** Direct Global WIS2 subscription

2.2 System Context Diagram¶



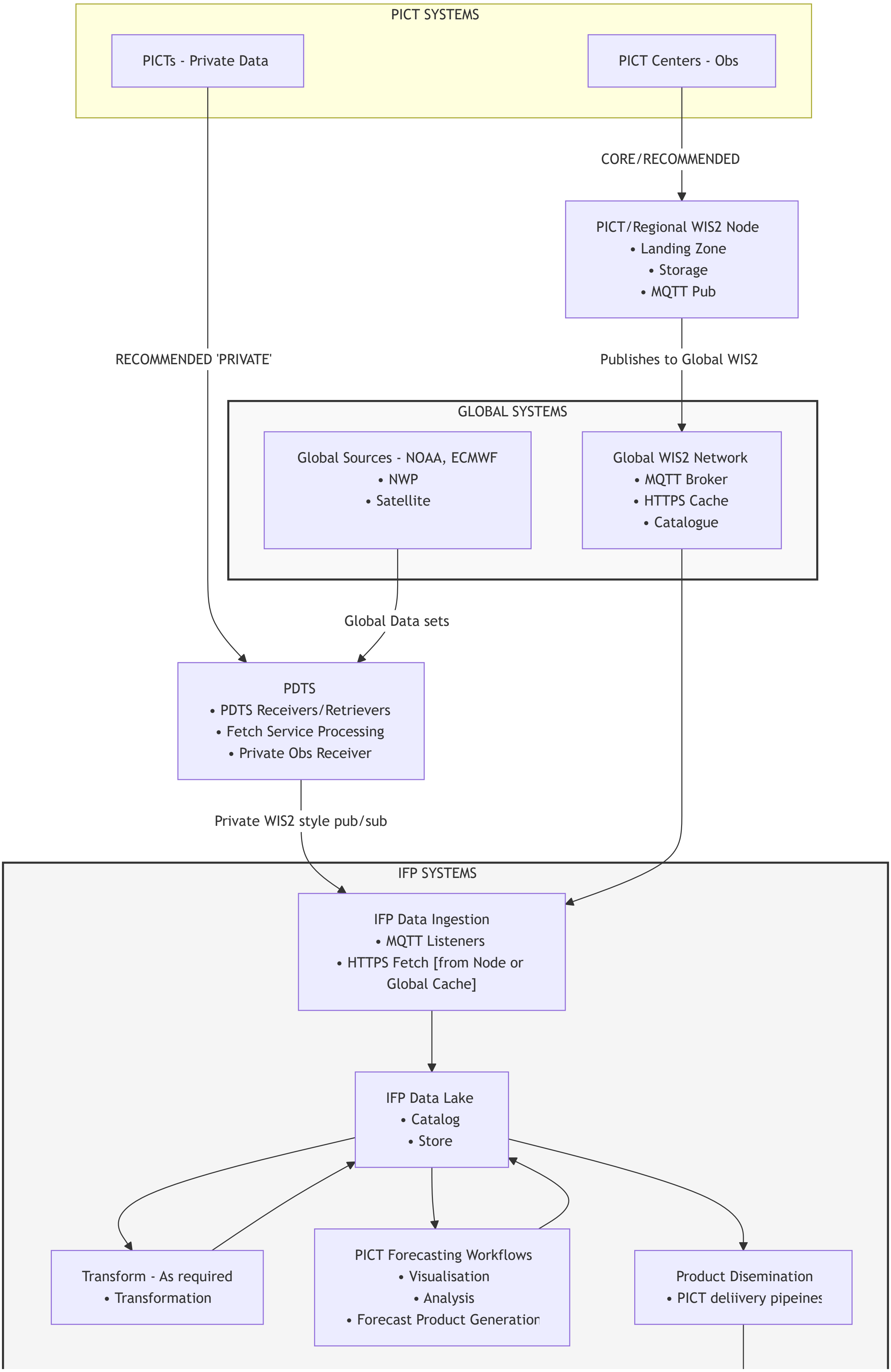
Colours indicate system ownership and responsibility:

- Global services (gold),
- National services - PICTs responsibility (green).
- Regional responsibility - IFP Project (light blue),
- Pacific IFP System - IFP Project (dark blue),

Figure 1: Complete Integrated Forecasting Platform Ecosystem

- Regionally sourced data that flows from PICT observation networks to the WIS2 global network either for Core/Recommended data sets, or to the Pacific Data Tailoring Service (PDTS) for private data sets.

- Globally sourced data such as NWP and Satellite imagery, is retrieved or subscribed to by the PDTS, optimised and made available to the IFP.
- The IFP then ingests from the Global WIS2 network or the PDTS and provides PICT based Forecaster workflows allowing the visualisation, analysis and generation of products that are then delivered out to operational consumer channels tailored to PICTs.


PICT SYSTEMS
PICTs - Private Data
PICT Centers - Obs
CORE/RECOMMENDED
PICT/Regional WIS2 Node
• Landing Zone
• Storage
• MQTT Pub
RECOMMENDED 'PRIVATE'
Publishes to Global WIS2
GLOBAL SYSTEMS
Global Sources - NOAA, ECMWF
• NWP
• Satellite
Global WIS2 Network
• MQTT Broker
• HTTPS Cache
• Catalogue
Global Data sets
PDTS
• PDTS Receivers/Retrievers
• Fetch Service Processing
• Private Obs Receiver
Private WIS2 style pub/sub
IFP SYSTEMS
IFP Data Ingestion
• MQTT Listeners
• HTTPS Fetch [from Node or Global Cache]
IFP Data Lake
• Catalog
• Store
Transform - As required
• Transformation
PICT Forecasting Workflows
• Visualisation
• Analysis
• Forecast Product Generation
Product Disemination
• PICT deliivery pipeines

2.3 Data Classification & Sovereignty Enforcement¶

The IFP and PDTS will ingest and store data sets based on the WMO WIS2 'topic-hierarchy' structure. By grouping data by PICT data centre id, this structure enables enforcement of data sovereignty across all systems. Data usage restrictions and enforcement are achieved through the Core and Recommended data classification policies. These classification policies also leave options open for future cost recovery implementations.

Data-Policy Classifications (Technical)¶

WIS 2.0 Data-Policy	Definition	Scope
Core	Global real-time Earth system data that is "essential for provision of services for the protection of life and property and for the well-being of all nations"	Essential WMO observations (mandatory)
Recommended	Additional data exchanged to "support broader Earth system monitoring" that falls outside the mandated essential umbrella of Core data	Globally Valuable data (with terms & conditions)
Recommended *	(*Private) PICT data restricted to Pacific regional IFP, and will not be catalogued, brokered or cached on the WMO WIS2 network	Required for PICT forecasting workflows (with terms & conditions)

* Private PICT data is classified as **RECOMMENDED** but kept private and is restricted to IFP via a **private delivery pipeline** using the Pacific Data Tailoring Service (PDTS) :

Delivery Pipeline	Data-Policy	Scope	Access	Publication
WIS2	Core	Global availability	Unrestricted (WIS2)	• PICT WIS2 Node or Regional Node, • Global WIS2 Catalog, Broker and Cache
WIS2	Recommended	PICT + approved users	Approved per PICT	• PICT WIS2 Node or Regional Node (gated), • Global WIS2 Catalogue and Broker only • Because Recommended data is often restricted, it is never mirrored by the public Global Caches
PDTS	Recommended	IFP-only (super-consumer)	IFP super-consumer (ADR-0001)	• **Private PDTS** hosted WIS2 Node

Key Distinction:

- **Data-Policy** (WIS 2.0): `core` or `recommended` only
- **Delivery Pipeline** (Operational): Determines WHERE data flows (Global WIS2, Regional WIS2, or PDTS private)
- **Private data:** Uses `recommended` data-policy but routed through PDTS private wis2box (never exposed to Regional WIS2 Hub Node or Global WIS2 Network)

Metadata Governance:

Usage restrictions: via metadata in WIS2 Discovery Catalogue metadata records:

```
properties [
 access_constraints: ["access-restricted"],
 licence: "proprietary",
 data_restrictions: "<bilateral agreement terms>",
]
```

Data access Enforcement Mechanisms are also documented in WIS2 Discovery Catalogue metadata records:

```
"security": {
  "default": {
    "description": "Please contact the data provider for access",
    "scheme": "bearer",
    "type": "http"
}
```

2.4 Retrieving Core and Recommended Data¶

When IFP consumes data through the WMO WIS2Box reference implementation, the technical mechanism for retrieving **Core** and **Recommended** data differs significantly. Understanding these differences is critical for designing robust data ingestion pipelines.

WMO 'CORE' Data Category¶

Definition: Global real-time Earth system data that is "essential for provision of services for the protection of life and property and for the well-being of all nations"

WIS2.0 Architecture Impact:

- Core Data **must be provided on a free and unrestricted basis**
- Automatically ingested by **Global Caches** (e.g., AWS S3 public buckets)
- Published via the `.../data/core/...` subtopic hierarchy
- **Access:** Unrestricted download via public HTTPS endpoints without authentication
- **Storage:** Mirrored to WMO Global Caches for high-availability distribution

Consumption Path:

1. IFP subscribes to MQTT topic: `origin/a/wis2/{node}/data/core/#`
2. Receives WIS2 Notification Message with `links[0].href` pointing to global cache
3. Downloads file via standard `curl` or `wget` (no authentication required)
4. Example: `https://wis2globalcache.s3.amazonaws.com/wis2/{node}/data/core/...`
   wis2/ca-eccc-msc/data/recommended/weather/surface-based-observations/2026-06-02-2212-CXKA-AUTO-minute-swob.xml

WMO 'RECOMMENDED' Data Category¶

Definition: Additional data exchanged to "support broader Earth system monitoring" that falls outside the mandated essential umbrella of Core data.

WIS2.0 Architecture Impact:

- Recommended Data supports broader Earth system monitoring but **can be subject to licensing or usage conditions**
- **NOT mirrored to Global Caches** - remains hosted solely on the originating Country Node
- Published via the `.../data/recommended/...` subtopic hierarchy
- **Access:** Restricted to approved consumers; may require authentication/authorisation
- **Storage:** Hosted on the originating wis2box node's web services (OGC APIs, direct file access)

Consumption Path:

1. **Connect to a WIS2 Broker :**

```
mqtt://{wis2-node}:1883
```

or the global WIS2 broker

```
mqtt://globalbroker.wmo.int:1883
```

1. **Subscribe to the Recommended Topic Filter:**

```
origin/a/wis2/{country-node}/data/recommended/weather#
```

1. **Receive WIS2 Notification Message** with properties:


```
{
  "id": "0f5b2c7c-b3ff-493a-a209-9bd8c0bd7364",
  "type": "Feature",
  "conformsTo": ["http://wis.wmo.int/spec/wnm/1/conf/core"],
  "geometry": null,
  "links": [
    {
      "href": "https://a-wis2-node.org/wis2/{centre-id}/data/recommended/weather/surface
      "length": 10421,
      "rel": "canonical",
      "security": {
        "default": {
          "description": "Please contact the data provider for access",
          "scheme": "bearer",
          "type": "http"
        }
      },"type": "application/xml"
    }
  ],
  "properties": {
    "data_id": "wis2/{centre-id}/data/recommended/weather/surface-based-observations/202
    "datetime": null,
    "global-cache": "sa-ncm-global-cache",
    "integrity": {
```

```
      "method": "sha512",
      "value": "39T+sYPE8hljwNnpi8EPKk2ixQzcfEDfcXECjByze7dc1JdBSwM3FEBmLixA6iUmyQt1oNJM
    },
    "metadata_id": "urn:wmo:md:{centre-id}:{dataset-identifier}", ## e.g. "urn:wmo:md:ca
    "pubtime": "2026-06-02T22:21:00Z"
  }  }
```

1. **Critical Payload Difference from Core Data:**

- `properties.data_id` explicitly contains `.../data/recommended/...` (not `.../data/core/...`)
- `links[0].href` points to **local country node** (not global cache)
- `metadata_id`: urn, use this to look up the metadata in a GDC using the urn to learn about the terms and conditions for usage e.g.:
- "https://wis2-gdc.weather.gc.ca/collections/wis2-discovery-metadata/items/urn:wmo:md:ca-eccc-msc:e50a9544-eee2-460c-a8b1-1a92a487d060",
- "https://gdc.wis.cma.cn/collections/wis2-discovery-metadata/items/urn:wmo:md:bn-bdmd:aviation-metar"
- `links.secuirty.scheme:` : *`bearer`* value indicates a bearer token is required to access the file

IFP Data Ingestion Strategy¶

The IFP should:

1. **Subscribe to Core and Recommended data via Global Broker** (unrestricted, high availability)
2. **Subscribe to Recommended ('private') data via Local Country Brokers** (only if explicitly approved)
3. **Use direct file downloads** from MQTT payload `href`: URL for both Core and Recommended
4. **Implement Bearer Token management** for Recommended data access (refresh tokens, error handling)
5. **Cache locally** to minimize re-downloads and reduce bandwidth consumption

---

2.5 Software Tools & Cost Profile¶

Purpose & Scope¶

The Integrated Forecasting Platform and supporting systems require a mix of open-source, licensed,
and managed cloud services. This section provides:

1. **Comprehensive tools inventory** organized by system (IFP, Regional WIS2 Node, PDTS) and deployment phase (Phase 1 Fiji on-premises, Phase 2+ AWS Sydney)
2. **Cost framework** organized into capital (one-time) and operating (recurring annual) categories
3. **Operational governance models** for three integrated systems that share common stewardship challenges

Strategic Architecture Decisions Reflected Here¶

The tools and cost profile reflect key architectural decisions documented elsewhere in this SAD:

- **Open-source strategy:** WIS2Box (WMO reference), WIS2downloader, RustFS minimize license burden
- **Managed cloud services:** AWS managed services (ECS/Fargate, S3, CloudWatch) reduce operational overhead
- **Phased deployment:** Phase 1 leverages existing Fiji investments (Visual Weather, on-premises DC);
  Phase 2 introduces cloud-native IFP (Open Weather, AWS Sydney)
- **Regional WIS2 Hub Node:** AWS deployment provides resilience and eliminates on-premises burden on remote centers
- **Unified governance model:** All three systems (WIS2 Node, PDTS, IFP AWS) require operational stewardship
  with similar roles and challenges

How to Read This Section¶

Tools Tables (Section 2.5.1): Each component lists:

- Which cost categories it maps to (can be multiple: capital purchase + operating evolution)
- Which phase(s) it applies to (Phase 1, Phase 2+, or both)
- Key notes (dependencies, why it's evolving over time)

Cost Framework (Section 2.5.2): 13 canonical cost categories (6 capital, 7 operating) used consistently
across all tools tables. A glossary and reference table are provided in the Appendix.

Operational Stewardship Models (Section 2.5.3): Three distinct governance contexts (WIS2 Node, PDTS, IFP AWS)
that must be defined to enable accurate operating cost estimates. The cost estimates in this section depend on
several outstanding decisions, including the operational governance structures (see Section 2.5.3).

Critical Path to Cost Estimation¶

Phase 1 Work

- AWS cost modeling (Regional WIS2 Node + PDTS compute/storage workload analysis)
- IBL licensing cost inquiry (Visual Weather + Online Weather capital and annual maintenance)
- Data storage volume analysis (1-year retention size estimate; critical S3 cost driver)
- Contractor RFP for engineering services (IFP + PDTS custom development)

Phase 2 Work

- Operational Stewardship governance models definition (roles, authority, staffing across three systems)
- Architecture finalization for IFP AWS Phase 2
- Fiji on-premises infrastructure scoping with in-country stakeholders

Blocking Items:

- Governance model decisions directly impact operating cost line items (Operational Stewardship FTE allocation)
- Data storage volume analysis critical for AWS Storage cost estimation
- Contractor RFP scope affects both Engineering capital costs and ongoing support budget

---

2.5.1 Software Tools by System¶

IFP (Integrated Forecasting Platform)¶

Phase 1: Fiji On-Premises Instance¶

Component	Tool/Service	Vendor	License	Cost Category(ies)	Pha
Backend - Server	Visual Weather (backend server)	IBL	Licensed	IBL Licensing, Operational Stewardship	1
Frontend - Client	Visual Weather Fat Client (user application)	IBL	Licensed	IBL Licensing	1
Data Ingestion - Global	WMO WIS2downloader	WMO	Open Source (AGPL)	Engineering (Initial Development)	1
	Alternative	IBL Moving Weather WIS2 Module	IBL	IBL Licensing	1
Data Ingestion - Regional	WRP-run regional NWP outputs	WRP/TBD	TBD	Engineering (Initial Configuration)	1
Orchestration	Python (custom)	Custom-built	N/A	Engineering (Initial Development), Operational Stewardship	1
Transform Service	Python (custom containers)	Custom-built	N/A	Engineering (Initial Development),	1

Component	Tool/Service	Vendor	License	Cost Category(ies)	Pha
				Operational Stewardship	
Product Dissemination Engine	Python (custom orchestrator + adapters)	Custom-built	N/A	Engineering (Initial Development, Initial Configuration), Operational Stewardship	1
Storage	RustFS (S3-compatible)	Open Source	Apache 2.0	Infrastructure Setup (Fiji), AWS Storage	1
Monitoring/Logging	CloudWatch or equivalent	AWS/TBD	Managed service	AWS Infrastructure	1
Infrastructure	Fiji Meteorological Service data center	On-premises	N/A	Infrastructure Setup (Fiji)	3

Phase 2: AWS Sydney Instance¶

Component	Tool/Service	Vendor	License	Cost Category(ies)	Pha
Backend - Server	Open Weather (backend server)	IBL	Licensed	IBL Licensing, Operational Stewardship	2+

Component	Tool/Service	Vendor	License	Cost Category(ies)	Pha
Frontend - Client	Online Weather SPA (user application)	IBL	Licensed	IBL Licensing	2+
Data Ingestion - Global	WMO WIS2downloader or IBL Moving Weather	WMO / IBL	Mixed	Engineering (Initial Development)	2+
Data Ingestion - Regional	WRP-run regional NWP outputs	WRP/TBD	TBD	Engineering (Initial Configuration)	2+
Orchestration	Python (custom, Lambda/ECS)	Custom-built	N/A	Engineering (Initial Development), Operational Stewardship	2+
Transform Service	Python (Lambda/ECS container)	Custom-built	N/A	Engineering (Initial Development), Operational Stewardship	2+
Product Dissemination Engine	Python (Lambda/ECS orchestrator + adapters)	Custom-built	N/A	Engineering (Initial Development, Initial Configuration), Operational Stewardship	2+

Component	Tool/Service	Vendor	License	Cost Category(ies)	Pha
Storage	AWS S3	AWS	Managed service	AWS Storage	2+
Monitoring/Logging	CloudWatch	AWS	Managed service	AWS Infrastructure	2+
Infrastructure	AWS ECS/Fargate (Sydney ap-southeast-2)	AWS	Managed service	Infrastructure Setup (AWS), AWS Infrastructure	2+

Regional WIS2 Hub Node¶

Deployment: AWS Sydney (ap-southeast-2)

Deployment Rationale: AWS Sydney deployment chosen for network resilience, compute elasticity, and ability to serve multiple PICT access patterns without increasing on-premises infrastructure burden on Fiji Data centre. Enables high-availability WIS2 node for Pacific region.

Component	Tool/Service	Vendor	License	Cost Category(ies)
Core Platform	WIS2Box	WMO	Open Source	Engineering (Initial Development), Operational Stewardship
Object Storage	MinIO (wis2box-minio fork)	WMO (maintained)	AGPL-3.0	Engineering (Initial Development)

Component	Tool/Service	Vendor	License	Cost Category(ies)
Config Management	Git version control	GitHub/GitLab/Self-hosted	Variable	Engineering (Initial Configuration), Git/DevOps
Dashboard/Admin	Grafana (built-in)	WIS2Box	Included	Engineering (Initial Development)
Bearer Token Management	WIS2Box admin tools (built-in)	WMO	Included	Operational Stewardship
Monitoring/Logging	CloudWatch	AWS	Managed service	AWS Infrastructure
Alerting/Notification	ICT/steward's service management system	TBD	TBD	Operational Stewardship
Infrastructure	AWS ECS/Fargate (Sydney)	AWS	Managed service	Infrastructure Setup (AWS), AWS Infrastructure

PDTS (Pacific Data Tailoring Service)¶

Deployment: AWS Sydney (ap-southeast-2)

Component	Tool/Service	Vendor	License	Cost Category(ies
Global Data Ingestion	WMO WIS2downloader	WMO	Open Source	Engineering (Initial Developmen
Custom Ingestion	Bespoke tools (polling/downloading)	Custom-built	N/A	Engineering (Initial Developmen Operational Stewardship
Geospatial Processing	Python (Lambda/ECS container)	Custom-built	N/A	Engineering (Initial Developmen Operational Stewardship
Transformation/Tailoring	Python (Lambda/ECS container)	Custom-built	N/A	Engineering (Initial Developmen Operational Stewardship
Orchestration	AWS S3 Events + SQS	AWS	Managed service	AWS Infrastructure
Storage	AWS S3	AWS	Managed service	AWS Storage
Monitoring/Logging	CloudWatch	AWS	Managed service	AWS Infrastructure

Component	Tool/Service	Vendor	License	Cost Category(ies
Alerting/Notification	ICT/stewards service management system	TBD	TBD	Operational Stewardship
Infrastructure	AWS ECS/Fargate (Sydney)	AWS	Managed service	Infrastructure Setup (AWS), AWS Infrastructure
Private WIS2Box	WIS2Box instance (IFP-only data)	WMO	Open Source	Engineering (Initial Development

---

2.5.2 Cost Framework¶

Cost Categories Glossary¶

These categories organize capital and operating costs across the three integrated systems.

Capital Costs (One-Time, Project Phase)¶

- **IBL Licensing (Capital):** Initial purchase of Visual Weather and/or Online Weather software licenses
- **Engineering (Initial Development):** Contractor and staff effort to design, build, and integrate custom platforms (Python orchestration, transform services, product dissemination orchestrator, bespoke PDTS tools)
- **Engineering (Initial Configuration):** Setting up initial workflows, data configurations, PICT routing rules, dissemination adapters, SOP documentation

- **Infrastructure Setup (Fiji):** On-premises server configuration, on-premises storage setup, networking
- **Infrastructure Setup (AWS):** VPC provisioning, security groups, networking, initial AWS resources configuration
- **Training & Change Management (Initial):** Forecaster training, stewards onboarding, SOP documentation, runbooks, internal knowledge transfer

Operating Costs (Recurring Annual, Steady-State)¶

- **IBL Licensing (Annual):** Maintenance, support, and license renewal fees for Visual Weather and/or Online Weather
- **AWS Infrastructure:** ECS/Fargate compute, data transfer, CloudWatch monitoring, managed service overhead
- **AWS Storage:** S3 storage tiers, backups, tiered retention dependent on requirements (e.g. 30-day active + 1-year archive), data transfer
- **Fiji On-Premises:** On-premises server maintenance, power, internet connectivity, local infrastructure support
- **Operational Stewardship:** Ongoing operations staff (FTE) for WIS2 Node stewards, PDTS operations team, IFP AWS operations team; includes governance, configuration management, incident response, training, and support for all three systems
- **Git/DevOps:** GitHub/GitLab subscriptions, CI/CD runners, version control infrastructure
- **Monitoring/Alerting:** CloudWatch, third-party SaaS monitoring tools, alerting infrastructure

Capital Costs (One-Time)¶

Cost Category	Components	Estimated Range	Status
IBL Licensing (Capital)	Visual Weather (Phase 1 Fiji) + Online Weather (Phase 2 AWS)	**TBD**	⚠️ To be confirmed with IBL; may be bundled or separate
Engineering (Initial Development)	IFP custom orchestration, transform services, product dissemination orchestrator; PDTS fetch coordinator, geospatial processing, transformation engine;	**TBD**	⚠️ Contractor RFP pending; Phase 1 + Phase 2 separate scopes

Cost Category	Components	Estimated Range	Status
	WIS2Box integration and customization		
Engineering (Initial Configuration)	Initial SOP workflows (Visual Weather), PICT routing configurations, dissemination adapters (country-specific), dataset configurations (WIS2Box), PDTS transformation rules, integration testing	**TBD**	⚠️ Depends on scope and forecaster input; Phase 1 first
Infrastructure Setup (Fiji)	Server hardware procurement, on-premises storage (RustFS) setup, networking configuration, containerization foundation	**TBD**	⚠️ To be scoped with Fiji IT; may be in-kind or funded
Infrastructure Setup (AWS)	VPC, security groups, networking, IAM roles, initial resource provisioning, DNS, storage buckets	**TBD**	⚠️ AWS architecture definition pending Phase 2 planning
Training & Change Management (Initial)	Forecaster training programs, stewards certification, operational documentation (runbooks, SOPs), change management communications	**TBD**	⚠️ To be scoped; critical for operational success

Subtotal Capital Costs: TBD (pending IBL quote, contractor RFP, infrastructure scope, and training plan definition)

Operating Costs (Recurring Annual)¶

Cost Category	Components	Estimated Range	Status
IBL Licensing (Annual)	Visual Weather + Online Weather maintenance,	**TBD**	TBD with IBL: The current estimate is

Cost Category	Components	Estimated Range	Status
	support contracts, vendor escalation		based on current Fiji implementation and assumes no additional VW modules required; in Phase 2 additonal licensing will need to be negationated with IBL for Open Weather/Online Weather implementation in AWS
AWS Infrastructure	ECS/Fargate compute for Regional WIS2 Node + PDTS + IFP AWS; CloudWatch monitoring; data transfer	**TBD**	Pending workload analysis; separate from storage costs
AWS Storage	S3 storage (active + archive), tiering, backups, data transfer	**TBD**	Data volume analysis needed; 1-year retention can be hundreds of TB
Fiji On-Premises	Server maintenance, OS patching, power, internet connectivity, local IT support	**TBD**	Depends on infrastructure approach and ISP agreement
Operational Stewardship	Staff FTE across all three systems: WIS2 Node stewards (regional data management, NMHS onboarding, bearer token management) + PDTS operations team (global data source management, transformation rule	**TBD**	**CRITICAL:** Governance models for all three systems TBD; significantly impacts total operating costs

Cost Category	Components	Estimated Range	Status
	evolution, integration support) + IFP AWS operations team (country workflow management, SOP evolution, forecaster training); includes governance, incident response, configuration evolution		
Git/DevOps	GitHub/GitLab subscriptions, CI/CD runners, infrastructure-as-code tooling	**TBD**	$2-5k/year if SaaS; $0 if self-hosted
Monitoring/Alerting	CloudWatch (AWS systems), third-party SaaS monitoring, alerting infrastructure	**TBD**	Part of AWS infrastructure costs or separate SaaS tool

Subtotal Operating Costs: TBD (pending AWS analysis, IBL maintenance confirmation, data volume analysis, and Operational Stewardship governance model definition)

2.5.3 Unified Operational Stewardship Model¶

Purpose & Scope¶

Operational Stewardship is the ongoing engineering and operations effort required to maintain, evolve, and support all three integrated systems (IFP-Fiji, IFP-AWS, Regional WIS2 Node, PDTS) in production.

Key Insight: While these systems have distinct architectures, they share common operational challenges:

- **Configuration evolution:** SOP workflows change, dissemination channels evolve, data sources need adjustment

- **Incident response:** Monitoring failures, data quality issues, integration breakdowns
- **Governance & decision-making:** Who approves what? How are conflicts resolved? How are policies enforced?
- **Staff continuity & training:** Maintaining expertise, onboarding new team members, knowledge transfer

Three Distinct Governance Contexts (to be defined in parallel):

WIS2 Node Stewardship (Regional Data Broker)¶

Responsibilities:

- Manage dataset configurations (dataset-configuration.yml) for participating NMHS
- Issue, renew, revoke bearer tokens for Recommended data access control
- Onboard new NMHS and define SLAs
- Monitor data quality and catch processing errors
- Respond to incidents (missing data, format errors, MQTT outages)
- Facilitate capacity planning and infrastructure management

Governance Questions (TBD):

- **Who are the stewards?** RSMC Fiji staff, dedicated regional team, SPREP-appointed committee, or hybrid?
- **Organizational ownership:** Who do they report to? Who has final authority on policies?
- **Decision authority:** Who approves data classifications (Core vs. Recommended)? Who resolves disputes between NMHS?
- **Training & expertise:** What WIS2 standards expertise is required? How is it maintained?
- **Escalation paths:** How are conflicts between NMHS or data policy disputes escalated and resolved?

Cost Impact:

- Staff FTE allocation (regional stewards team; estimated 1-2 FTE equivalent)
- Training & certification programs (WIS2 standards, operational procedures)
- Support infrastructure/tools (Git, monitoring, alerting, bearer token management)

Status: ⚠️ Governance structure **TBD before Phase 1 pilot launch**

PDTS Operations (Global Data Tailoring & Private Data)¶

Responsibilities:

- Select and prioritize global data sources (NOAA GFS, ECMWF, JMA, satellite, etc.)
- Manage data ingestion schedule and fallback strategies
- Develop and maintain transformation rules (cropping, compression, format conversion)
- Ingest PICT private observations and publish via private WIS2Box
- Monitor data quality and integration health
- Provide first-line support to IFP teams on data issues
- Scale infrastructure as data volumes grow

Governance Questions (TBD):

- **Who owns PDTS operations?** RSMC Fiji staff, external contractor, shared responsibility, or managed service?
- **Data source decisions:** Who decides which global models to ingest? Forecasters? Steering committee?
- **Transformation rule management:** Who maintains and approves changes to cropping parameters, compression settings, format conversions?
- **Integration support:** Who troubleshoots IFP/WIS2 integration issues caused by PDTS data problems?
- **Private data handling:** Who defines which PICT observations are "private" vs. "public"? Who manages access?
- **Escalation paths:** How are data quality issues escalated? Who decides if a data source is unreliable?

Cost Impact:

- Staff FTE allocation (PDTS operations team; estimated 1-2 FTE equivalent)
- Training & onboarding programs (geospatial processing, data source management)
- Operational monitoring infrastructure
- Integration testing/validation tools

Status: ⚠️ Operations model **TBD before Phase 2 design finalization**

IFP AWS Operations (Phase 2+ Distributed Forecasting)¶

Responsibilities:

- Manage country-specific product dissemination workflows and adapters

- Support SOP evolution as forecasting procedures change
- Monitor forecaster access and user experience
- Respond to incidents (system failures, data delays, integration issues)
- Train remote National NMHS Forecasters on new features/workflows
- Scale infrastructure as forecaster adoption grows

Governance Questions (TBD):

- **Who owns IFP AWS operations?** RSMC Fiji, shared with PICT IT teams, AWS managed service, or distributed team?
- **Workflow changes:** Who approves changes to forecasting procedures or product types?
- **Dissemination adapters:** Who develops/maintains adapters for new PICT channels (CMS, social media, aviation APIs, etc.)?
- **Forecaster support:** Who provides first-line support to remote forecasters using Online Weather SPA?
- **Training & onboarding:** Who trains new forecasters in each PICT? How is competency verified?
- **Escalation paths:** Who handles critical incidents? Who is on-call?

Cost Impact:

- Staff FTE allocation (IFP AWS operations team; estimated 1-2 FTE equivalent, likely shared with other systems)
- Training & onboarding programs (forecaster certification, new workflow procedures)
- Support infrastructure/tools (help desk, ticketing, monitoring)
- Documentation (runbooks, troubleshooting guides, SOP updates)

Status: ⚠️ Operations model **TBD during Phase 2 planning**

Impact on Cost Estimation:

- Operational Stewardship FTE and budget allocation
- Training and onboarding programs
- Support infrastructure and tools

2.5.4 Summary Table for Stakeholders¶

System	Key Software Tools	License Cost	Infrastructure Cost	Engineering Development (One-off)	Ong Stew Cost
IFP (Fiji)	Visual Weather backend + Fat Client (licensed) + WIS2downloader (open) + Python (custom)	TBD (IBL quote pending)	$0 (existing DC)	TBD (Phase 1 development team)	TBD (I alloca pendi
IFP (AWS)	Open Weather backend + Online Weather SPA (licensed) + WIS2downloader (open) + Python (custom)	TBD (IBL quote pending)	TBD (AWS pay-as-you-go)	TBD (Phase 2 development team)	TBD (I alloca pendi
Regional WIS2 Node	WIS2Box (open) + MinIO (open) + Git (open)	$0	TBD (AWS pay-as-you-go)	TBD (pilot implementation)	TBD (stewa config token manag
PDTS	WIS2downloader (open) + Python (custom) + AWS services	$0	TBD (AWS pay-as-you-go)	TBD (Phase 2 development team)	TBD (curati pipeli maint
Operational Stewardship (Unified)	Governance, incident response, configuration evolution, training across all systems	$0	Included above	N/A (operational cost only)	**TBD (gover mode pendi**
TOTAL	**Hybrid (open + licensed +**	**TBD (IBL quote**	**TBD (AWS)**	**TBD**	**TBD (gove**

System	Key Software Tools	License Cost	Infrastructure Cost	Engineering Development (One-off)	Ong Stewa Cost
	managed)	**pending)**			**pendi**

Section 3: Container Views & System Internals¶

3.0 Deployment Architecture: Two Independent IFP Instances¶

Overview¶

The **Integrated Forecasting Platform (IFP)** is deployed as **two independent instances** operating in parallel:

1. **Fiji IFP (On-Premises)** - RSMC Fiji regional hub
2. **AWS Sydney IFP (Cloud)** - Remote PICT distributed platform

Each instance is **fully self-contained** with its own data lake, orchestrators, and dissemination engines. They operate independently by default, with optional product sharing (currently TC advisories only).

Instance 1: Fiji IFP (On-Premises) - Phase 1 - IFP Project¶

Location: Fiji Meteorological Service data center (on-premises infrastructure)

Purpose: Authoritative regional forecasting hub; serves as primary source for regional TC advisories and aviation products

Users: RSMC Fiji forecasters (Visual Weather Fat Client)

Full Container Stack:

- Data Ingestion Worker (WIS2 + PDTS subscription)
- Data Transformation Engine
- Visual Weather Server (+ Fat Client for Fiji forecasters)
- TC Module (TC advisory generation)
- Product Dissemination Engine → routes to PICT adapters (including Fiji)

- PICT Dissemination Adapters (includes Fiji and all remote PICTs: Samoa, Tonga, Kiribati, Vanuatu, Solomon Islands, etc.)

Data Lake: Three S3 buckets equivalents (on-premises using RustFS):

- `ifp_object_store_bucket` (ingested raw data)
- `ifp_products_bucket` (generated forecasts, 7-day retention)
- `ifp_archive_bucket` (historical products)

Infrastructure: On-premises server(s), can be supplemented with private cloud (AWS, Azure, etc.) for storage/compute

Instance 2: AWS Sydney IFP (Cloud) - Phase 2 - IFP Project¶

Location: AWS ap-southeast-2 region (Sydney)

Purpose: Distributed forecasting platform for remote PICTs; each country's forecasters use lightweight web interface

Users: National NMHS Forecasters (Open Weather SPA web client)

Full Container Stack:

- Data Ingestion Worker (WIS2 + PDTS subscription)
- Data Transformation Engine
- Open Weather Server (+ Online Weather SPA for PICT web access)
- TC Module (optional; can consume Fiji TC advisories)
- Product Dissemination Engine → routes to PICT adapters (including Fiji)
- PICT Dissemination Adapters (includes Fiji and all remote PICTs: Samoa, Tonga, Kiribati, Vanuatu, Solomon Islands, etc. - one adapter handles all PICTs' channels)

Data Lake: Three S3 buckets (AWS managed):

- `ifp_object_store_bucket` (ingested raw data)
- `ifp_products_bucket` (generated forecasts, 7-day retention)
- `ifp_archive_bucket` (historical products)

Infrastructure: AWS ECS/Fargate or similar container orchestration; fully managed cloud service

Deployment Architecture Decisions¶

Independence (Default):

- Each instance operates completely independently
- Fiji forecasts products do NOT automatically route to AWS adapters
- AWS forecasts products do NOT automatically route to Fiji adapters
- Each instance manages its own region's dissemination pipeline
- Separate orchestrators, separate adapters, separate data lakes

Product Sharing (Exception - TC Advisories to start with):

- Fiji-generated TC advisories are shared with AWS IFP
- **Flow:** Fiji Dissemination Engine has special TC Adapter → writes TC products to AWS IFP `ifp_products_bucket`
- AWS IFP ingests Fiji TC advisories as if they were just-created products
- AWS forecasters see Fiji TC advisories alongside their own products
- **Implementation:** Fiji IFP to AWS_Cloud IFP delivery adapter (for shared products)
- **Future Extensibility:** Other product types (marine warnings, public forecasts) could be added later if needed

Infrastructure Choices:

- **Fiji:** On-premises for sovereignty and control; can grow to hybrid (on-prem + private cloud)
- **AWS:** Cloud-native for scalability and reduced on-premises infrastructure burden on remote PICTs
- **No Direct Integration:** Instances do NOT share databases or queues; each has isolated infrastructure

Scaling & Resilience:

- Each instance can scale independently based on demand
- AWS instance naturally scales via ECS/Fargate auto-scaling
- Fiji instance scales via on-premises capacity planning or hybrid cloud
- Failure in one instance does NOT impact the other
- Fiji TC sharing is optional; if cross-instance adapter fails, each instance continues independently

User Experience¶

Fiji Forecasters (On-Premises Instance):

- Use Visual Weather Fat Client (rich desktop application)

- Generate Forecast products including TC advisories (authoritative regional source)
- Products disseminate to Fiji-specific channels (CMS, social media, email, SMS)
- TC Advisories disseminated to AWS IFP for regional sharing

Remote National NMHS Forecaster (AWS Instance):

- Use Open Weather SPA web client (lightweight browser-based interface)
- Access regional PICT data lake and forecasts
- Can see and consume Fiji-generated TC advisories (optional, via cross-instance sharing)
- Products disseminate to PICT-specific channels (social media, email, local systems, WIS2)

Deployment Timeline¶

Phase 1 (Current):

- Fiji IFP (on-premises) operational
- TC advisory sharing enabled (Fiji → AWS)

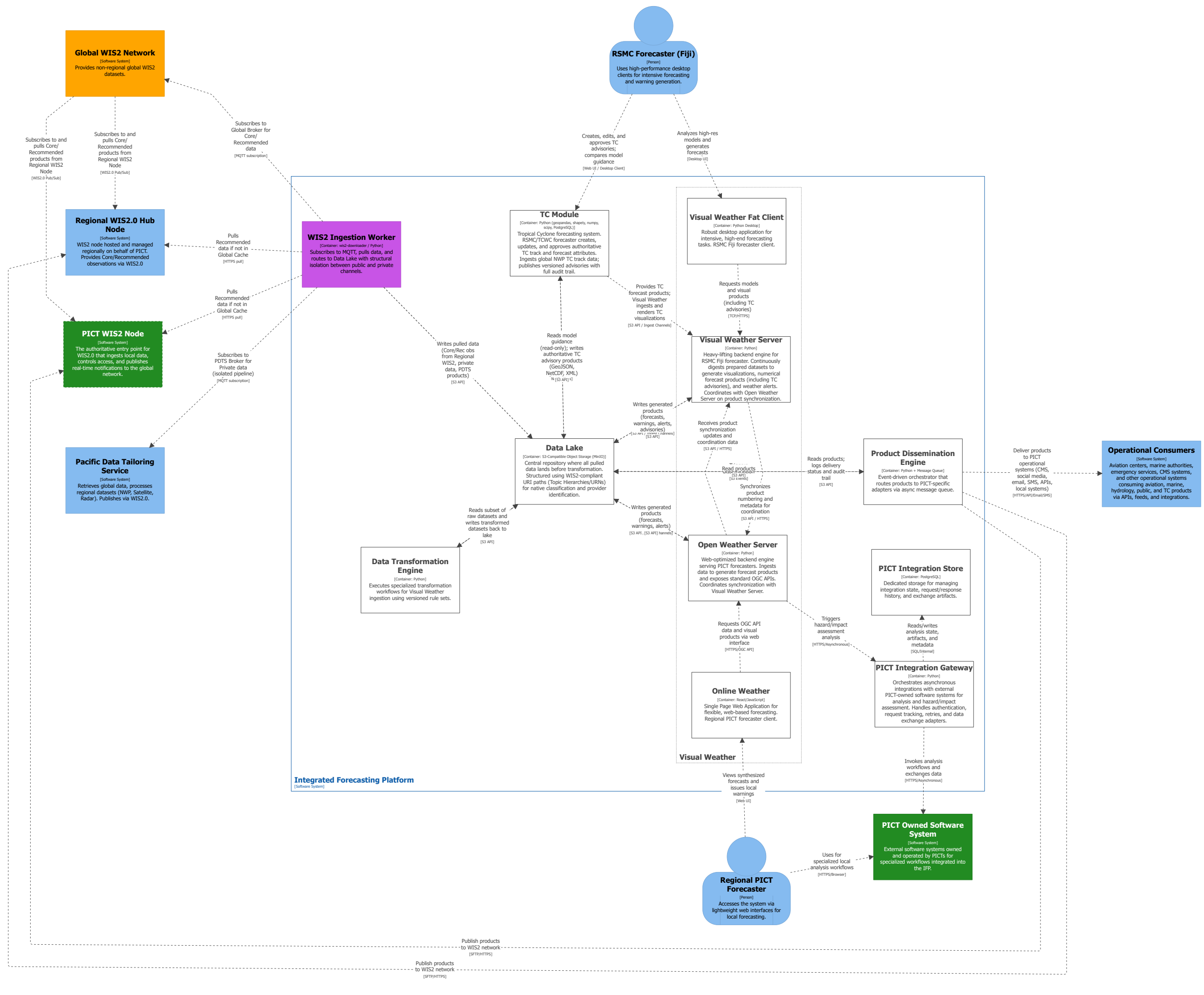
Phase 2+ (Future):

- AWS IFP (cloud) operational
- Extend product sharing to other product types (marine, public forecasts)

3.1 IFP Container View (PRIMARY SYSTEM)¶

Overview¶

The **Integrated Forecasting Platform (IFP)** comprises **7 major containers** that work together to ingest data, transform formats, generate forecasts, visualize for forecasters, and disseminate products.



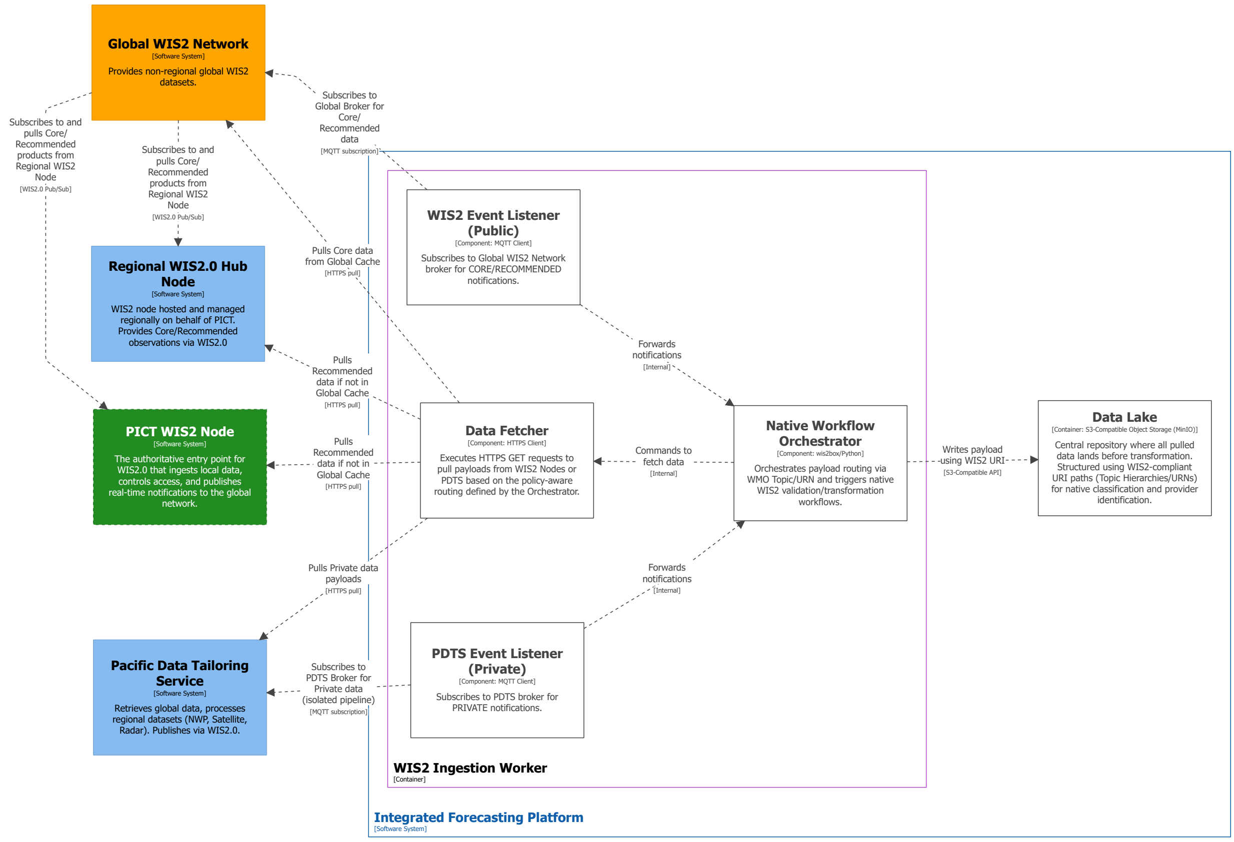
Container View: Integrated Forecasting Platform
Thursday, June 11, 2026 at 4:34 PM New Zealand Standard Time

Components¶

3.1.1 Data Ingestion Worker¶

Responsibility: Connect to Global WIS2 Network and PDTS private wis2box; fetch and ingest meteorological observations, NWP models, and satellite imagery directly into `ifp_object_store_bucket`.

No landing zone: Data flows directly from source → Data Fetcher → `ifp_object_store_bucket`, eliminating intermediate staging.



Component View: Integrated Forecasting Platform - WIS2 Ingestion Worker
Component-level view of the WIS2 Ingestion Worker.
Thursday, June 11, 2026 at 4:01 PM New Zealand Standard Time

Architecture: Two-Component Pipeline

a) Native Workflow Orchestrator (Python)

- **Responsibility:** Route MQTT notifications and determine fetch strategy
- **Inputs:**
- MQTT notifications from Global WIS2 Broker (CORE/RECOMMENDED data)
- MQTT notifications from PDTS private broker (RECOMMENDED data via private pipeline)
- **Outputs:** Fetch commands to Data Fetcher
- **Key Functions:**
- Subscribe to relevant MQTT topics (based on WIS2 topic hierarchy)
- Parse WIS2 Notification Messages for payload URLs and metadata
- Determine access method (Global Cache, WIS2 Node origin, PDTS origin) based on data-policy and message descriptors
- Route fetch commands to Data Fetcher

Implementation: Can leverage WMO WIS2downloader tools for MQTT subscription and message parsing, or implement custom Python components

b) Data Fetcher (HTTPS Client)

- **Responsibility:** Execute fetch operations and store directly to Data Lake
- **Inputs:** Fetch commands (URL, storage path, token reference if RECOMMENDED data)
- **Outputs:** Data objects in `ifp_object_store_bucket` (using WIS2 topic hierarchy path)
- **Key Functions:**
- Execute HTTPS GET requests with appropriate authentication
- **Core Data:** No authentication required (download from Global Cache)
- **Recommended Data:** Include bearer token in Authorisation header (tokens from secure configuration)
- Store payloads directly to S3 using WIS2 topic hierarchy path
- Handle retries, timeouts, partial downloads
- Verify integrity (checksum if available)
- No format conversion or validation (stored as-is)
- **Token Management:** Bearer tokens stored in secure configuration; mapped to specific PICT WIS2 nodes (see Section 2.2.5 for token management model)

Connection Management:

- MQTT connection state monitoring (reconnect on disconnect)
- Backpressure handling (queue management if fetches fall behind)
- Rate limiting per source (respect WIS2 node quotas)

Technologies: MQTT client library, HTTPS client, S3 API client (boto3), Python framework (optionally using WMO WIS2downloader tools for MQTT message handling)

ADR References:

- `IFP/decisions/0002` (WIS2 Ingestion Mechanism)
- `IFP/decisions/0003` (Asynchronous Data Integration)
- `IFP/decisions/0005` (Dual Data Ingestion Sources)

3.1.2 Data Lake (S3-Compatible Object Storage with RustFS)¶

Responsibility: Persistent storage for meteorological datasets and forecast products.

Logical Concept vs. Physical Implementation:
The **Data Lake** is a logical concept representing the complete meteorological data repository. It is implemented as multiple S3-compatible buckets, each serving a distinct purpose in the data lifecycle.

Interfaces:

- **Input:** Data objects are stored in the data lake by:
- Data Ingestion Worker (direct ingest from WIS2/PDTS)
- Data Transformation Engine (transformed outputs)
- TC Module (TC advisories and related products)
- Forecast Generation Engine (forecast objects)
- **Output:** Data lake objects are accessed by:
- Visual Weather (visualisation)
- TC Module (forecast inputs)
- Product Dissemination Pipelines (product generation)
- PICT Integration Gateway (per-country filtering)
- Archive Manager (retention-based archival)
- **Persistence:** Per-bucket retention policies; archive zones for historical data

Physical Bucket Structure:

Bucket Name	Purpose	Retention	Access	Trigger Events
`ifp_object_store_bucket`	Active operational data (real-time to current month)	30 days	All operational components	S3 bucket events → Transformation Orchestrator
`ifp_archive_bucket`	Historical data for analysis and model training	1+ years	Research and validation	None (read-only archive)

Data Partitioning Strategy:

Both buckets use the **WIS2 topic hierarchy** to organize data by source, data-policy, and type. This ensures:

- Automatic classification preservation (core/recommended)
- Consistent routing for ingestion, transformation, and archival

- Simplified access control (path-based governance)

WIS2 Topic Hierarchy Path:

```
wis2/{data-centre-id}/data/{data-policy}/{topic-short-name}/{object-identifier}
```

Example: Surface Observations (PICT Data)

Ingested from WIS2:

```
ifp_object_store_bucket/wis2/fj-rsmc/data/core/weather/surface-observations/2026-06-10-221
```

After Transformation (BUFR → NetCDF):

```
ifp_object_store_bucket/wis2/fj-rsmc/data/core/weather/surface-observations/2026-06-10-221
```

(Data-policy preserved; format changed only)

After Archive (30+ days old):

```
ifp_archive_bucket/wis2/fj-rsmc/data/core/weather/surface-observations/2026-06-10-2212-CXK
```

Technologies: AWS S3-compatible API with S3 Bucket event notification feature.

Implementation: RustFS - RustFS | The High-Performance, Apache 2.0 Alternative to MinIO

Why RustFS (not MinIO)?

- ✅ **100% native S3 API parity** - drop-in replacement for S3-compatible workflows
- ✅ **S3 bucket event notifications** - critical for triggering downstream transformation workflows
- ✅ **Apache 2.0 license** - no licensing constraints (unlike MinIO's BUSL model)
- ⚠️ **Technology choice rationale:** MinIO is reserved for the WIS2 Node implementations because the WMO WIS2Box reference was implemented with it. For IFP and PDTS in which we will be implementing S3-compatible storage, RustFS has been selected to avoid MinIO licensing overhead.

S3 Bucket Event Integration:

RustFS S3 bucket events notify the **Transformation Orchestrator** when new objects land in `ifp_object_store_bucket`. Event payloads include object key and metadata, triggering the transformation pipeline in an event-driven manner (no polling required).

Related Containers:

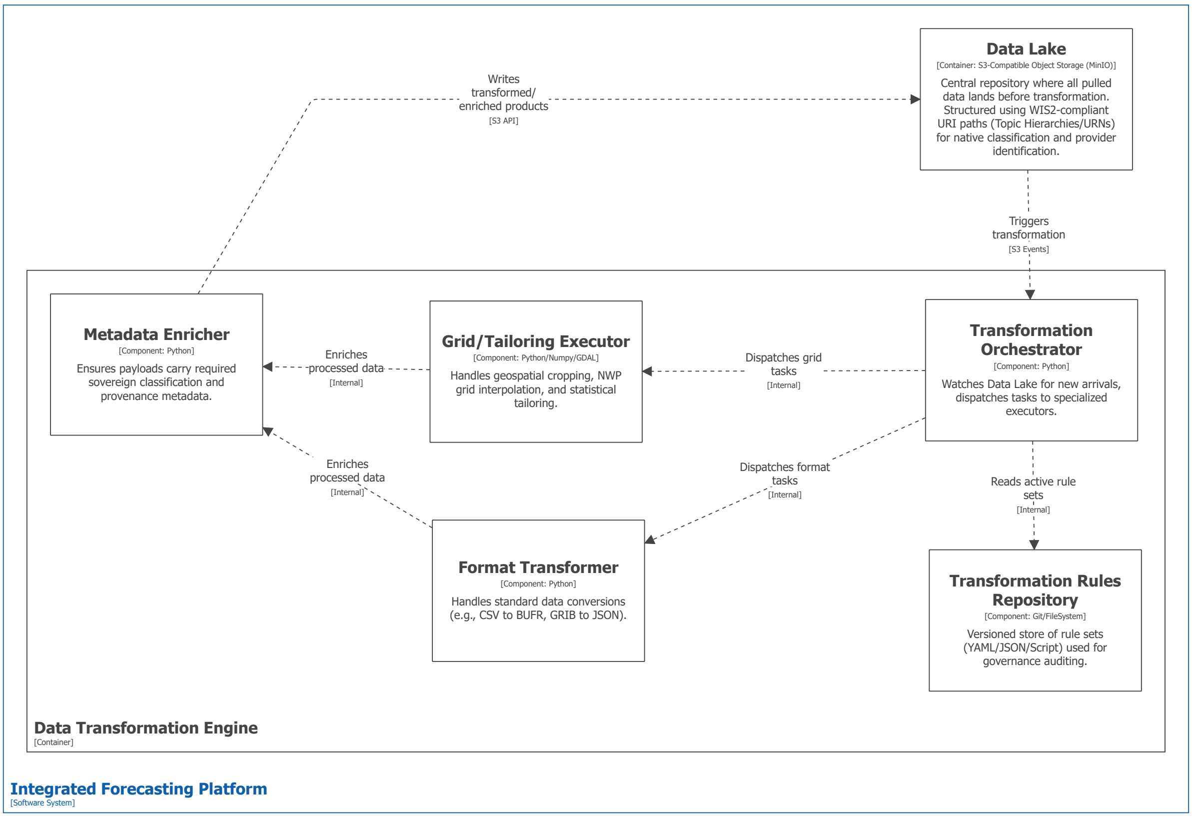
- **Transformation Orchestrator** - subscribes to S3 bucket events
- **Archive Manager** - moves objects from active bucket to archive bucket based on retention policy

ADR References:

- `IFP/decisions/0013` (Data Lake Bucket Architecture)
- `Regional_WIS2Node/decisions/0006` (S3-Compatible Storage as Edge Ingestion)

3.1.3 Data Transformation Engine¶

Responsibility: Execute specialized transformation workflows for data ingestion using versioned rule sets. Transform ingested data into standardized formats optimized for operational forecasting and visualisation.



Component View: Integrated Forecasting Platform - Data Transformation Engine
Component-level view of the Data Transformation Engine.
Thursday, June 11, 2026 at 4:01 PM New Zealand Standard Time

Container-Level Overview:

The **Data Transformation Engine** is a single container with five specialized sub-components:

a) Transformation Orchestrator (Python)

- **Responsibility:** Route S3 bucket events and determine which transformations to apply
- **Inputs:** S3 bucket events from `ifp_object_store_bucket` (object arrived notifications)
- **Outputs:** Dispatch commands to appropriate executors
- **Key Functions:**
- Subscribe to RustFS S3 bucket events
- Parse object key to extract WIS2 topic hierarchy
- Identify data-policy (core/recommended) and data type (surface-obs, NWP, satellite, radar, etc.)
- Look up transformation rules from Transformation Rules Repository
- Dispatch to appropriate executor (Grid/Tailoring, Format Transformer, or Metadata Enricher)

b) Transformation Rules Repository (Git/FileSystem)

- **Responsibility:** Versioned store of rule sets for governance and auditability
- **Storage:** YAML/JSON/Script files in Git or filesystem
- **Content:** Mapping rules (e.g., "BUFR → NetCDF", "GRIB2 → NetCDF", surface-obs transformation rules, NWP grid cropping parameters)
- **Key characteristic:** Versioned (enables governance auditing and rollback)

c) Grid/Tailoring Executor (Python/Numpy/GDAL)

- **Responsibility:** Handle geospatial and grid-based transformations
- **Transforms:** NWP grid cropping, interpolation, statistical tailoring, satellite compositing
- **Use cases:** Pacific region optimisation, vertical interpolation (model levels → pressure levels)
- **Output format:** NetCDF4 with zlib compression or GeoTIFF (Cloud Optimized)

d) Format Transformer (Python)

- **Responsibility:** Handle standard data format conversions
- **Transforms:** BUFR ↔ NetCDF, GRIB2 ↔ NetCDF, CSV ↔ BUFR, etc.
- **Libraries:** netCDF4, pygrib, custom parsers
- **Output:** Standardized formats (NetCDF preferred for observations and model data)

e) Metadata Enricher (Python)

- **Responsibility:** Ensure transformed objects carry required metadata
- **Functions:**
- Add/verify data-policy classification (core/recommended)

- Add provenance metadata (source, transformation timestamp, lineage)
- Add sovereign attribution (PICT centre-id)
- Ensure WIS2 topic hierarchy compliance

Workflow:

1. S3 bucket event arrives (object landed in `ifp_object_store_bucket`)
2. Transformation Orchestrator receives event with object key (e.g., `wis2/fj-rsmc/data/core/weather/surface-observations/2026-06-10-synop.bufr`)
3. Parses WIS2 topic hierarchy to identify:
   - Data-policy: `core` (from path)
   - Data type: `surface-observations` (from path)
   - Format: `bufr` (from file extension)
4. Looks up transformation rules from Transformation Rules Repository:
   - Rule: "surface-observations/BUFR → NetCDF conversion + metadata tagging"
5. Dispatches to appropriate executor:
   - Format Transformer (BUFR → NetCDF)
6. Executor processes data and passes to Metadata Enricher
7. Metadata Enricher adds:
   - Data-policy preservation: `core` (unchanged from input)
   - Provenance: transformation timestamp, source lineage
   - Attribution: centre-id `fj-rsmc`
8. Enricher writes transformed object back to `ifp_object_store_bucket` using **same WIS2 topic hierarchy path** with updated format:
   - `ifp_object_store_bucket/wis2/fj-rsmc/data/core/weather/surface-observations/2026-06-10-synop.netcdf`

Key Design Principles:

- ✅ **Data-Policy Preservation:** Transformed objects retain original classification (core/recommended)
- ✅ **Path Consistency:** WIS2 topic hierarchy used for input routing, output storage, and archival
- ✅ **Event-Driven:** S3 events trigger orchestrator (no polling or tight coupling)
- ✅ **Format Standardisation:** Conversions to NetCDF or domain-specific standards (GeoTIFF for satellite)
- ✅ **Governance Auditable:** Rule sets versioned in Git for compliance and rollback capability

- ✅ **Metadata Aware:** All outputs carry sovereign attribution and lineage tracking

Technologies:

- **Orchestrator & Enricher:** Python
- **Grid/Tailoring:** Python, Numpy, GDAL, xarray
- **Format Transformer:** Python, netCDF4, pygrib, custom parsers
- **Rules Repository:** Git or filesystem (YAML/JSON/Python scripts)
- **Storage interaction:** S3 API (boto3)

Related Containers:

- **Data Lake** (S3-compatible RustFS) - S3 events trigger orchestrator; transformed outputs written back
- **Forecast Generation Engine** - Consumes transformed data from Data Lake
- **Product Dissemination** - Transforms forecasts into CAP, RSS, etc.

ADR References:

- `IFP/decisions/0016` (WIS2 Native Ingestion and Validation)
- `IFP/decisions/0014` (Data Transformation Architecture)

3.1.4 Archive Manager¶

Responsibility: Manage data lifecycle by moving aged data from active bucket to archive.

Interfaces:

- **Input:** Objects in `ifp_object_store_bucket` (monitoring retention policy)
- **Output:** Moved/copied objects in `ifp_archive_bucket`
- **Triggers:** Scheduled (daily or on-demand) based on retention policy

Workflow:

1. Query `ifp_object_store_bucket` for objects older than retention threshold (e.g., 30 days)
2. Copy object to `ifp_archive_bucket` using **same WIS2 topic hierarchy path**
3. Optionally compress for long-term storage efficiency (preserve path and metadata)
4. Verify integrity (checksum, metadata preservation)
5. Delete from active bucket (or mark as archived)

Retention Policy:

- **Active bucket:** 30 days (operational forecasting)
- **Archive bucket:** 1+ years (research, model training, historical analysis)

Technologies: S3 lifecycle policies or custom Python/Go orchestration with boto3/AWS SDK

3.1.5 Forecast Generation Engine¶

Responsibility: Provide forecasters with visualisation and analysis tools to generate meteorological forecast products. Supports both manual forecaster workflows and automated forecast generation triggered by data ingestion. Forecast generation workflows are configured within the Visual Weather and Open Weather products, consuming data from the Data Lake and storing forecast products back to the Data Lake.

Dual-Mode Operation:

- **Manual Mode:** Forecasters use VW Client/Server or Open Weather SPA to analyze data and author forecast products
- **Automated Mode:** Data ingestion into Data Lake triggers configured FGE workflows to automatically generate forecast products without forecaster intervention

TC Forecast Workflow (Integration with TC_Module):

1. S3 events trigger TC_Module when TC track data arrives in Data Lake
2. RSMC Forecaster analyzes and approves TC forecast within TC_Module
3. TC_Module stores raw TC forecast data to Data Lake
4. Forecaster then uses VW Client/Server to review TC data and generate CAP alerts and final TC forecast products

Interfaces:

- **Input:** Data Lake (observations, NWP models, satellite, TC forecast data - from `ifp_object_store_bucket`)
- **Output:** Forecast objects (weather forecasts, TC products, decision-support products) stored back to Data Lake
- **Triggers:** Manual (forecaster action via VW Client/Server) and event-driven (data arrival triggers automated workflows)

Asynchronous Processing: Data arrival triggers event-driven forecast workflows. No polling or tight coupling to data sources.

ADR References:

- `IFP/decisions/0006` (TC_Module as Separate Container)
- `IFP/decisions/0008` (TC CAP Manual Workflow)
- `IFP/decisions/0012` (Dual Backend Architecture)

Visualisation & Analysis Subsystems (Dual Backend):

The Forecast Generation Engine provides forecasters with visual tools to analyze data and create/edit forecasts through **two distinct backends optimized for different personas**, all consuming the same unified Data Lake.

Design Rationale:

- RSMC Forecaster (Fiji) needs high-performance desktop tool for intensive analysis
- National NMHS Forecaster (remote) need lightweight web access
- Different personas require different client environments
- Shared Data Lake enables both backends to consume same data

---

3.1.5.1 Visual Weather Fat Client (RSMC Backend)¶

Purpose: High-performance forecasting workbench for RSMC Forecaster (Fiji)

Client: Visual Weather Fat Client (proprietary desktop application)

Features:

- Advanced meteorological visualisation (animated maps, cross-sections, etc.)
- Direct access to all data (Core, Recommended)
- Interactive forecast editing tools
- TC advisory creation and approval workflow
- CAP alert composition and dispatch
- Product quality review and coordination

Technical Stack:

- Proprietary binary protocol (optimized for performance)
- High-performance graphics rendering (GPU-optimized)
- Real-time data streaming

- Supports on-premises Fiji deployment

Personas Served: RSMC Forecaster

Data Access: All classifications (Core, Recommended) from both Global WIS2 and PDTS private wis2box

ADR Reference: `IFP/decisions/0012` (Dual Backend Architecture)

3.1.5.2 Visual Weather Backend Server (RSMC Backend)¶

Purpose: Backend server providing proprietary APIs for Visual Weather Fat Client

Features:

- Proprietary binary protocol optimized for performance
- Real-time data streaming to Fat Client
- Handles TC advisory and CAP integration
- Product quality review coordination
- Caching for high-performance access
- Direct access to all data (Core, Recommended)

Technical Stack:

- Proprietary binary protocol (optimized for performance over standard REST/JSON)
- Direct S3 access to Data Lake (boto3)
- In-memory caching (Redis or similar)
- On-premises deployment (Fiji LAN)

Personas Served: RSMC Forecaster

Data Access: All classifications (Core, Recommended) from both Global WIS2 and PDTS private wis2box

ADR Reference: `IFP/decisions/0012` (Dual Backend Architecture)

3.1.5.3 Open Weather Backend Server (PICT Backend) - Phase 2 of IFP Project¶

Purpose: Big Data weather database and Web Service engine that publishes weather data through open standard OGC protocols

Features:

- OGC API standards compliance (WCS - Web Coverage Service, EDR - Environmental Data Retrieval API)
- Big Data weather database for storing meteorological data
- Web Service engine publishing data via open standards
- Server-side filtering and aggregation (reduce bandwidth)
- Real-time data access and historical archival
- Seamless data exchange and accessibility via open standards

Technical Stack:

- OGC standards: WCS (Web Coverage Service) and EDR API (Environmental Data Retrieval)
- Big Data database backend (handles volume and velocity of meteorological data)
- Web Service engine for publishing via open protocols
- Server-side filtering and aggregation (reduce bandwidth)
- Open standards protocols for interoperability
- Cloud-deployable (AWS ECS or on-premises)

Personas Served: National NMHS Forecaster (via Online Weather frontend)

Data Access: Core/Recommended observations from Global WIS2; limited access to optimized global data via RSMC-filtered channel

Key Characteristics:

- Optimized for internet access (not LAN)
- Open standards (WCS, EDR API) instead of proprietary protocol
- Promotes interoperability and integration with external systems
- Enables external PICTs and third-party systems to access weather data
- Decoupled from user interface (consumed by Online Weather SPA)

Design Rationale:

By adhering to open protocols (WCS and EDR API), Open Weather provides a standardized framework for publishing and accessing diverse weather data. This promotes interoperability, allowing users to easily integrate weather information into their systems and applications, fostering innovation and collaboration within the meteorological community.

ADR Reference: `IFP/decisions/0012` (Dual Backend Architecture)

3.1.5.4 Online Weather (PICT Client)¶

Purpose: Web-based forecasting software designed to empower informed decision-making and streamline planning for National NMHS Forecaster

Client: Online Weather Web Application (browser-based single-page application)

Features:

- No software to install - this is a web application aka a single-page application (SPA) that runs entirely in the browser, consuming data from the Open Weather Backend Server via OGC APIs (WCS, EDR API)
- User-centred interface for accessible decision-making
- Interactive map-based visualisation
- Forecast browsing and interpretation
- Alert reception and acknowledgment
- Local forecast annotation and collaboration toolsMQTT ACL language a
- Mobile-friendly responsive design
- Template-driven forecast workflows
- Streamlined planning and product generation

Technical Stack:

- HTML5, JavaScript, WebGL for modern, performant rendering
- REST/JSON consumption of Open Weather backend APIs (WCS, EDR API)
- Progressive asset loading (minimize initial payload)
- Responsive design (desktop, tablet, mobile)
- Offline-capable (local caching)
- User-centric interface design for ease of use

Personas Served: National NMHS Forecaster

Network Context: Internet access (6-110 Mbps across PICTs, median ~40-50 Mbps); cannot support fat client deployment

Data Backend: Consumes Open Weather backend server APIs (WCS, EDR API)

Why Separate from RSMC Fat Client?

- Fat Client not suitable for remote internet access (latency, bandwidth)
- Web-based software provides lightweight access to authorized subsets
- Modern web technologies (HTML5, JavaScript, WebGL) enable responsive UX
- Reduced training burden for multi-disciplinary forecasters

- Progressive loading tolerates higher latency
- Mobile compatibility supports multiple devices per forecaster

ADR Reference: `IFP/decisions/0012` (Dual Backend Architecture)

3.1.6 Product Dissemination Engine¶

Responsibility: Event-driven dissemination of PICT forecast products to PICT-specific delivery channels with per-country configuration and channel-specific adapters.

Architecture: The Product Dissemination Engine consists of two layers:

Layer 1 - Core Orchestrator (Lightweight Routing)

- Receives S3 events when products are written to `ifp_products_bucket`
- Parses product metadata using WIS2 topic hierarchy (extracts PICT ID and product type)
- Loads PICT-specific routing configuration from version control
- Determines which adapters should handle the product
- Dispatches routing event to message queue (SQS/RabbitMQ/Kafka/TBD)
- Logs all dissemination events for audit trail

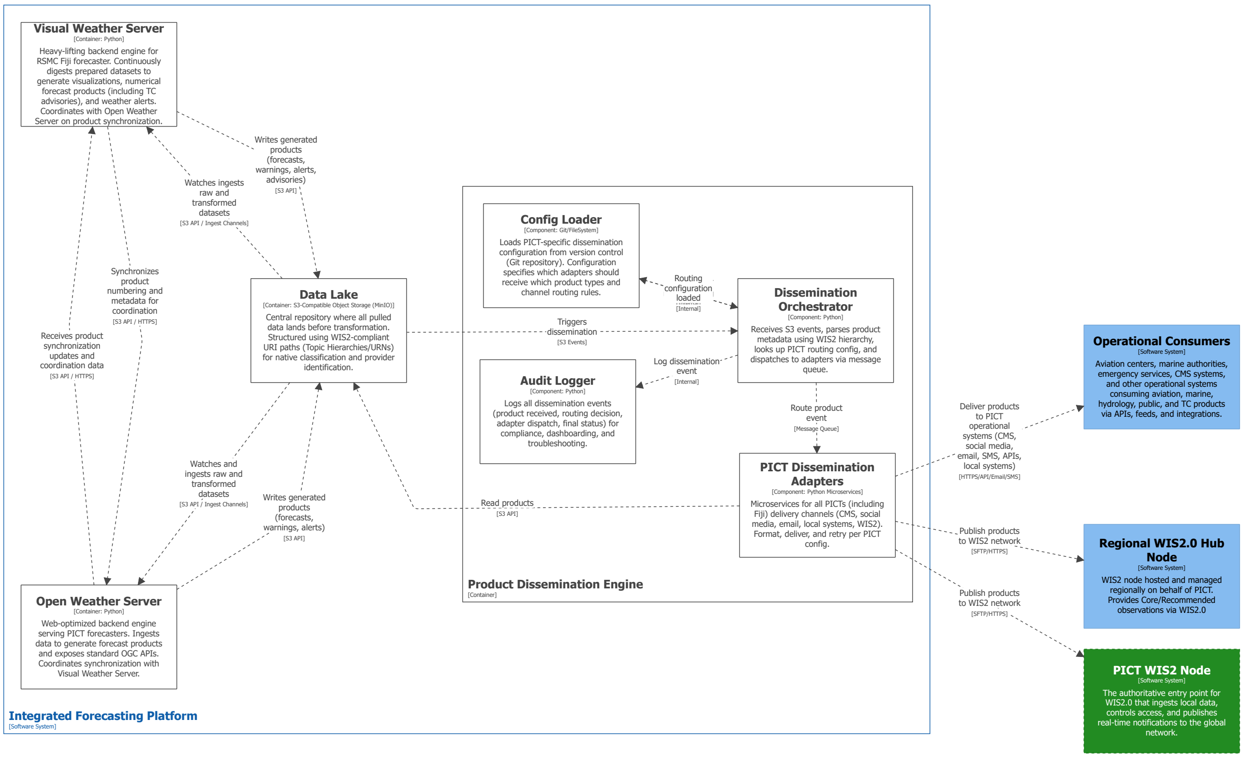
Layer 2 - PICT-Specific Adapters (Channel-Specific Delivery)

- **PICT Dissemination Adapters:** Single microservice component for all PICTs (including Fiji) delivery channels (CMS, social media, email, local systems, WIS2). Format, deliver, and retry per PICT configuration.

Each adapter:

- Consumes routing events from message queue (asynchronously)
- Reads product from `ifp_products_bucket` (S3 API)
- Applies channel-specific format transformation (if needed)
- Delivers to PICT's configured delivery channels
- Handles retry logic with exponential backoff
- Alerts operators on persistent delivery failures
- Reports delivery status back to core engine for audit logging

Interfaces:



Component View: Integrated Forecasting Platform - Product Dissemination Engine
Component-level view of the Product Dissemination Engine (core orchestrator only; adapters are separate containers).
Thursday, June 11, 2026 at 4:34 PM New Zealand Standard Time

- **Input:** Products written to `ifp_products_bucket` by Visual Weather Server, Open Weather Server, or TC Module
- **Output:** Disseminated products via PICT delivery channels:
- **WIS2 Publication:** SFTP/HTTPS to Regional WIS2 Hub Node (regional/global distribution)
- **Operational Consumers:** HTTPS/API, Email/SMS to PICT-specific operational systems (aviation centers, marine services, emergency management, CMS, social media, local government)
- **Triggering:** S3 object events (event-driven, no polling)
- **Routing:** Configuration-driven (externally managed via version control)

Key Functions:

- **Event-Driven Trigger:** S3 object events trigger immediate product routing (no latency from polling)
- **PICT-Centric Routing:** Each PICT controls which channels receive which product types
- **Configuration-Based:** Routing defined in external configs; integrating new channel requires config + adapter code (not core engine changes)
- **Asynchronous Dispatch:** Adapters consume routing events independently via message queue (decoupled from orchestrator)

- **Format Transformation:** Adapters handle channel-specific format conversion (product → CMS format, → social media text, → CAP XML, → aviation briefing, etc.)
- **Retry & Alerting:** Adapters own retry logic, alerting, and status reporting
- **Audit Trail:** All dissemination decisions logged for compliance and dashboarding

PICT Configuration (Version Control):
Example configuration for Fiji:

```yaml
pict_id: fj-rsmc
adapters:
  - fiji_dissemination_adapters
routing:
  - product_type: aviation
    channels:
      - cms_webhook
      - social_media
      - wis2_publication
  - product_type: marine
    channels:
      - email_alert
      - wis2_publication
  - product_type: tc_advisory
    channels:
      - cms_webhook
      - wis2_publication
      - operationalConsumers
```

ADR References:

- `IFP/decisions/0011` (Product Dissemination Container Architecture - Revised 2026-06-11)
- `IFP/decisions/0010` (Visual Weather CAP Generation)
- `IFP/decisions/0009` (TC Module Async Processing)
- `IFP/decisions/0016` (WIS2 Native Ingestion - S3 Events pattern)

Design Proposal:

- `IFP/designs/20260611_01` (S3 Event-Triggered Product Dissemination)
- `IFP/designs/20260611_02` (Product Dissemination Grilling Session Summary)

3.1.7 PICT Integration Gateway (Future Phase 2+)¶

Responsibility: Allow for PICT forecast workflows in the IFP to integrate with PICT software systems.

Interfaces:

- **Input:** Master IFP Data Lake queries
- **Output:** Per-PICT filtered data subsets

Key Functions:

- Per-PICT authorisation (enforce data sovereignty at instance level)
- Country-specific data filtering (each PICT sees only their data + authorized regional/global)
- Governance policy enforcement (enforce PICT-specific rules)

When Used: Phase 2+ when PICTs have operational capacity for dedicated instances

Status: Design in progress; not part of Phase 1 scope

3.2 Regional WIS2.0 Hub Node Container View (SUPPORTING SYSTEM)¶

Overview¶

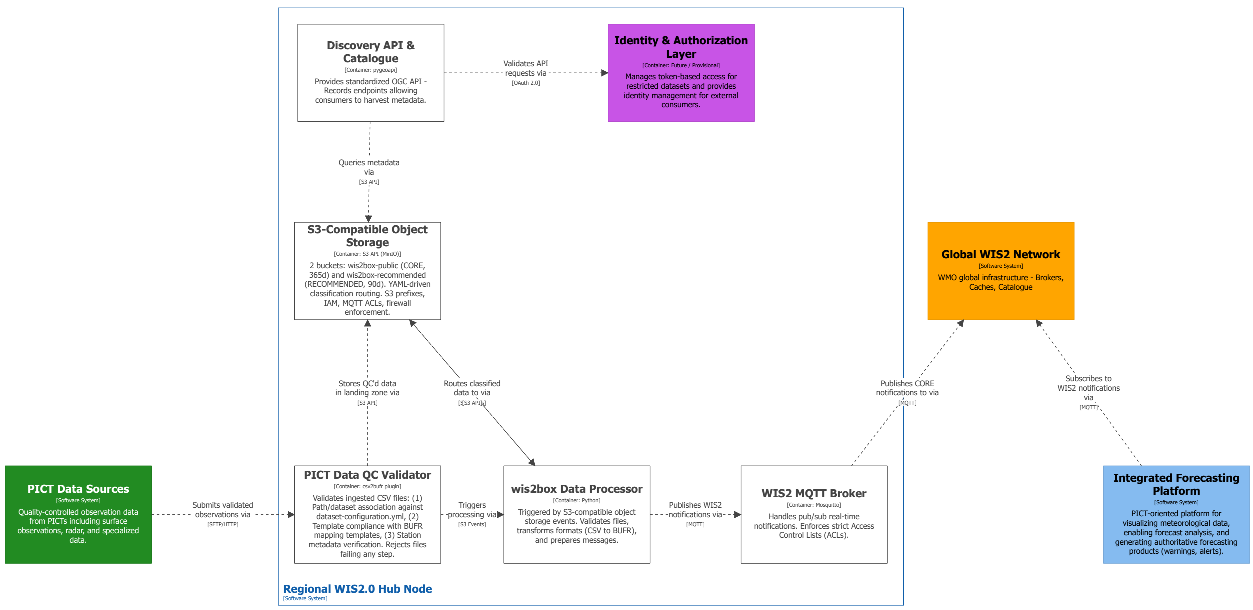
The **Regional WIS2.0 Hub Node** is a shared WIS2 node resource (deployed as PATH 2 Model A: single shared wis2box in Fiji or AWS Cloud Sydney). It ingests quality-controlled data sets supplied by Pacific Island Countries and Territories (PICTs), classifies them as CORE or RECOMMENDED, and publishes them to the Global WIS2 Network.

Purpose: Where a PICT is not able to implement their own WIS2 Node, the Regional Hub provides a means to get PICT data into the Global WIS2 Network (and by extension, to the IFP via standard WIS2 subscription). All PICT data routed through the Regional Hub is published to the **Global WIS2 Network** (not directly to IFP); IFP consumes this data like any other WIS2 subscriber.

⚠️ **IMPLEMENTATION NOTE:** This section describes the *internal container architecture of standard wis2box* deployed in AWS Sydney as PATH 2 Model A (shared regional hub). **We are deploying standard wis2box software as-is; we are NOT building custom replacements for these components.** The containers described below (MinIO, Data QC Validator, Data Processor, MQTT Broker, Discovery API, etc.) are provided by the wis2box project. Specific implementation details and deployment configuration will be refined based on Synoptic Data's wis2box implementation plan (led by Max Marno).

Critical Architectural Principles (per ADRs):

- ✅ **Hub Scope:** CORE + RECOMMENDED (public) data destined for Global WIS2 Network (using WIS2 topic hierarchy for path organisation and data sovereignty enforcement)
- ✅ **Data Sovereignty:** Achieved through the WMO WIS2 'topic hierarchy' structure. By grouping data by PICT data centre id, this structure enables enforcement of data sovereignty within the shared WIS2 Node
- ✅ **Pre-Ingestion QC:** PICTs perform Stage 1 QC before submission; Regional Hub validates during ingestion and halts processing if CSV→BUFR conversion fails
- ✅ **Fail-Fast Validation:** If validation or transformation fails, file is rejected and error is logged; no MQTT publication occurs; administrators see failures via Grafana dashboard
- ✅ **Event-Driven Processing:** All transformation workflows are triggered by data arrival events (S3 events); no polling or manual triggering required



Container View: Regional WIS2.0 Hub Node
Wednesday, May 27, 2026 at 2:05 PM New Zealand Standard Time

Data Sovereignty Architecture: Three-Phase Flow¶

The Regional WIS2 Hub Node implements data sovereignty through WIS2 topic hierarchy and path-based isolation. This flow describes how NMHS data flows through the system while preserving individual national attribution and control:

Phase 1: Ingestion & Validation (Securing Data Sovereignty)

1. **Isolated Raw Data Uploads:**
   - Regional stewards provision **dedicated, isolated ingest folder paths** for each participating NMHS
   - Example: `incoming/nmhs-fiji/` , `incoming/nmhs-samoa/` , `incoming/nmhs-tonga/`
   - Each NMHS uploads observations to their dedicated path (via SFTP or S3 API)
   - Clear operational boundaries; prevents cross-contamination of data or station IDs

2. **NMHS-Specific Event Validation:**
   - When a file lands, automated validator triggers
   - Validator cross-references the **file's folder path** against central `dataset-configuration.yml`
   - Maps each isolated ingest path to that specific NMHS's {data-center-id}
   - Validates file syntax and cross-references data against that NMHS's **authorized station metadata list**
   - Prevents data from one country being misattributed to another; enforces country-specific station ID validation

3. **Processing & Internal Publication:**
   - If validation passes, Data Processor transforms file (CSV → BUFR, as needed)
   - Writes transformed data to regional node's **public-facing storage** (`wis2box-public`)
   - While physical storage is shared, data is stored in directory structure reflecting originating NMHS
   - Path structure: `{nmhs-centre-id}/data/{core|recommended}/{topic}/{filename}`

Phase 2: Notification & Global Routing (Preserving Identity)

4. **Sovereign MQTT Notification Generation:**
   - Regional node generates WIS2 Notification Message (JSON payload with direct HTTPS download link)
   - **Critical step for data sovereignty:** Constructs WIS2 Topic Hierarchy using originating country's ID:
   `origin/a/wis2/{nmhs-centre-id}/data/{core|recommended}/{topic}`
   - Example: `origin/a/wis2/fj-rsmc/data/core/weather/surface-observations`
   - This MQTT topic guarantees global network attributes data to specific NMHS, not regional host

5. **The Routing Split (CORE vs. RECOMMENDED):**
   - **Path A (CORE Data):**
     - Regional node publishes notification to **WMO Global Broker** (public relay)

- Global Cache (AWS) receives notification, instantly fetches file from regional node, mirrors into high-speed cloud storage
- End-users subscribe to Global Broker; download mirrored file from Global Cache (anonymously, no auth required)
- Regional node is source-of-truth; Global Cache is performance mirror
- **Path B (RECOMMENDED Data):**
- Notifications remain on **regional MQTT broker** under NMHS's specific topic path
- Global Cache ignores (does NOT mirror)
- External users must subscribe to regional broker and download directly from regional node
- Download requires **bearer token** (manually issued by regional stewards; configured per NMHS/dataset)
- NMHS maintains full control over who can access restricted/licensed data

---

Phase 3: Metadata Discovery & Programmatic Access (Shared Infrastructure, Individual Attribution)

6. **Metadata Hosting (OGC API - Records) - Phase 1:**
   - Regional stewards manage single centralized **OGC API - Records endpoint** (shared infrastructure)
   - However, WCMP2/GeoJSON metadata records **explicitly declare the respective NMHS as data owner and author**
   - Example record:
   `json { "properties": { "title": "Fiji Meteorological Service Surface Observations", "data_center_id": "fj-rsmc", "author": "Fiji Meteorological Service" } }`
   - **Entirely public and unauthenticated** (WIS2 principle: metadata discovery is unrestricted)
   - Enables global search engines and users to discover PICT datasets while preserving NMHS attribution
7. **Global Catalogue Harvesting:**
   - WMO Global Discovery Catalogue **periodically harvests** Records API endpoint
   - Pulls metadata for ALL hosted NMHSs from single regional API
   - Harvesting is efficient (batch operation, not real-time)
   - Metadata indexed globally while **preserving individual national attributions**
   - Users worldwide can search global index and discover PICT data sources

8. **Targeted Data Retrieval (OGC API - EDR/Features) - Phase 2+:**
   - Power users or automated systems (like IFP) can query regional node's **OGC API - EDR** (future capability)
   - Access control: CORE data public; RECOMMENDED data requires API key/bearer token

---

Outstanding Governance Issue: Regional Stewards Role Definition¶

⚠️ **CRITICAL DECISION PENDING:** The operational success of this shared infrastructure depends on clearly defining the **regional stewards** role and structure.

Proposed Governance Model (to be tabled to Directors for decision)¶

The following governance allocation has been proposed for the MOU and requires Directors' approval:

Role	Organization	Key Responsibilities	Cost/FTE Implications
Operational IT & Incident Response	RSMC Fiji	System administration, infrastructure management, incident response, 24/7 on-call coverage	Remunerated for regional responsibilities; FTE allocation TBD
Data Configuration & Token Management	SPREP	Dataset configuration (dataset-configuration.yml), bearer token issuance/renewal, data policy management, SOPs	Part of stewardship FTE (estimated 0.5 FTE)
NMHS Onboarding & Training	SPREP	New NMHS onboarding, training programs, capacity building across region	Part of stewardship FTE (estimated 0.5 FTE)
Ongoing Cost Payment	SPREP	AWS infrastructure costs, license maintenance, operational tooling	Annual operating budget TBD
Regional Backup Capability	Participating NMHSs	Train 2 staff per NMHS to provide backup capability pool; support RSMC Fiji during peak events	Remunerated for regional responsibilities on activation

MOU Requirements:

- Address outstanding governance issues raised above
- Define remuneration structure for regional responsibilities (Fiji and trained NMHS backup staff)
- Specify escalation paths and decision authority
- Document SLAs for bearer token issuance, NMHS onboarding, and incident response
- Table to Directors for formal decision

Currently Unresolved (pending Directors' decision):

- **Remuneration structure:** How will Fiji and backup NMHS staff be compensated for regional duties?
- **Reporting authority:** Do stewards report to RSMC Fiji Director, SPREP, or regional committee?
- **Escalation paths:** How are disputes between NMHS data policies resolved?
- **Backup activation:** Under what conditions are NMHS backup staff activated? What is the mobilization protocol?

Action Items:

- [ ] Draft MOU with proposed governance allocation above
- [ ] Table governance model to Directors for decision
- [ ] Define remuneration structure for regional responsibilities
- [ ] Document escalation paths and decision authority
- [ ] Create Stewards Role Definition Document (post-approval)
- [ ] Update this SAD section once governance model is confirmed

Related ADRs:

- `Regional_WIS2Node/decisions/0008-cloud-platform-and-operational-ownership.md` (governance context)

Deployment & Operational Context¶

Deployment Model (Phase 1-2): PATH 2, Model A (Single Shared wis2box)

- **Location:** AWS ap-southeast-2 (Sydney) or Fiji data center (deployment location TBD per governance decision)

- **Operated By:** RSMC Fiji staff (with regional stewards supporting data configuration and bearer token management)
- **Scaling:** Can scale from single instance (Phase 1) to multiple regional instances (Phase 2+) without architectural changes

Regional Hub as Backup for National Nodes (PATH 1 Countries):

The Regional WIS2 Hub Node can serve as a backup data source for National WIS2 Nodes operated by PATH 1 countries. This is a **future capability** to be designed in Phase 2+:

Scenario	Regional Hub Role	Implementation
National Node failure (PATH 1 country)	Regional Hub continues publishing that country's data if configured as secondary ingestion point	Phase 2+ design: Dual publication pathway; National Node primary, Regional Hub secondary
Regional Hub failure (Model A)	AWS multi-AZ deployment provides infrastructure resilience; no automatic National Node fallback	Phase 1: Monitoring, alerting, incident response by stewards; Phase 2+: Evaluate Model B for isolation
Both Hub and National Node down	Global WIS2 caches provide limited fallback for CORE data	Design consideration: WIS2 global cache TTL and subscription resilience

Current Status: The Regional Hub is the **primary** platform for PATH 2 countries (no National Node exists for them). For PATH 1 countries, Regional Hub backup capability is **not yet designed** and requires:

- [ ] Architectural design for dual ingestion pathways
- [ ] Governance agreement between Regional Hub operators and PATH 1 NMHS
- [ ] Data synchronization and conflict resolution procedures
- [ ] SLA definition for backup activation and failover

Phase 2+ Recommendation: Evaluate Regional Hub backup role for PATH 1 countries based on operational experience and stakeholder requirements.

OGC API Implementation Status:

- ✅ **Phase 2 (2026-2027):** OGC API - Records (metadata discovery endpoint; entirely public)

- ⏳ **Phase 2+ (2027+):** OGC API - Features & EDR (power user data retrieval; requires API keys for RECOMMENDED data)

Technologies Summary:

Component	Technology	Purpose
Object Storage	S3-compatible (MinIO)	NMHS data landing zone and public file storage
Data Validation	Python + csv2bufr	Validate ingested files; fail-fast on errors
Data Transformation	Python + BUFR libraries	Transform CSV → BUFR; preserve metadata
MQTT Broker	Mosquitto	Publish WIS2 notification messages globally
Metadata API	pygeoapi	OGC API - Records (Phase 1); Features/EDR (Phase 2+)
Configuration Management	YAML + dataset-config	Stewards manage dataset definitions, bearer tokens
Monitoring & Administration	Grafana + Prometheus	Operational metrics, error tracking, admin dashboard

ADR References:

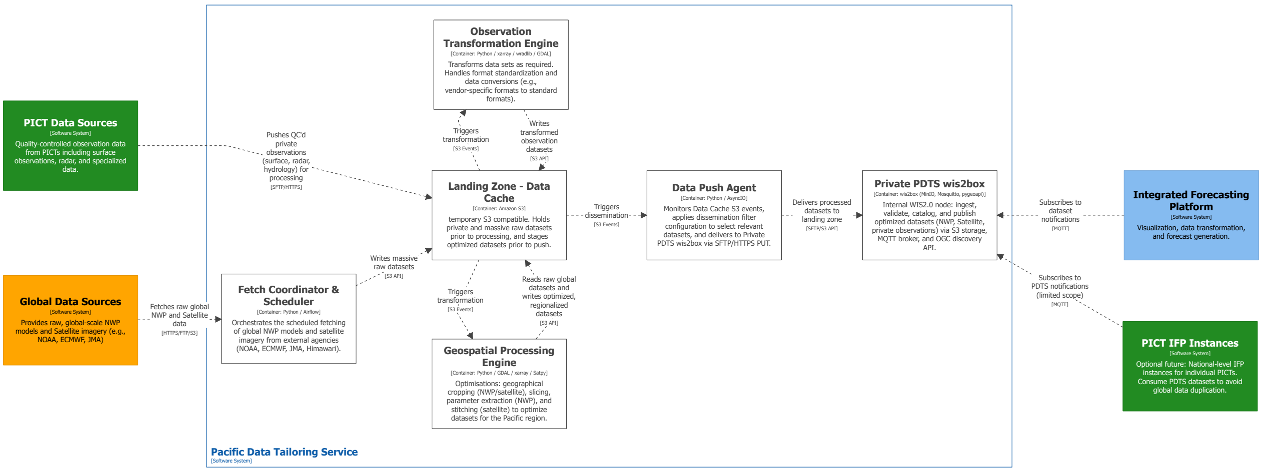
- `Regional_WIS2Node/decisions/0016-deployment-model-options-shared-vs-multi-node.md` (Model A rationale)
- `Regional_WIS2Node/decisions/0008-cloud-platform-and-operational-ownership.md` (cloud platform & governance)

3.3 Pacific Data Tailoring Service (PDTS) Container View (SUPPORTING SYSTEM)¶

Overview¶

The key roles of the Pacific Data Tailoring Service (PDTS) is Global data retrieval, geospatial optimisation and private PICT data routing. The PDTS retrieves massive global

NWP and satellite datasets, geospatially optimizes them for the Pacific region , ingests PICT private data sets, and then catalogs and publishes these dataset exclusively for the IFP via a private
wis2box implementation



Container View: Pacific Data Tailoring Service
Tuesday, 9 June 2026 at 6:33 pm New Zealand Standard Time

As with the IFP the PDTS will ingest and store data sets based on the WMO WIS2 'topic-hierarchy' structure. By grouping data by PICT data centre id, this structure enables enforcement of data sovereignty across all systems. Data usage restrictions and enforcement are achieved through the Core and Recommended data classification policies. These classification policies also leave options open for furture cost recovery implementations.

Components¶

3.3.1 Fetch Coordinator & Scheduler¶

Responsibility: Orchestrate scheduled retrieval of global NWP and satellite data.

Example Schedule:

- **GFS:** Every 6 hours (4 cycles/day)
- **ECMWF:** Every 12 hours (2 cycles/day)
- **JMA:** Every 12 hours (2 cycles/day)
- **Access-G (Australia):** Every 24 hours
- **Satellite:** Every 10 minutes to hourly (varies by source; Himawari-8 typically every 10-30 min)
- **TC tracks:** Every 6 hours or event-driven

Data Sources & Protocols:

- **NOAA:** HTTPS/FTP bulk downloads or NOAA-cloud (AWS S3)
- **ECMWF:** HTTPS API with authentication
- **JMA:** HTTPS mirror or sFTP
- **TC data:** WIS2 subscriptions or direct APIs

Failure Handling:

- Retry logic (exponential backoff)
- Notification on repeated failures
- Fallback to secondary sources (if available)

3.3.2 Landing Zone - Data Cache¶

Responsibility: temporary storage for Data ingestion and transformation.

Characteristics:

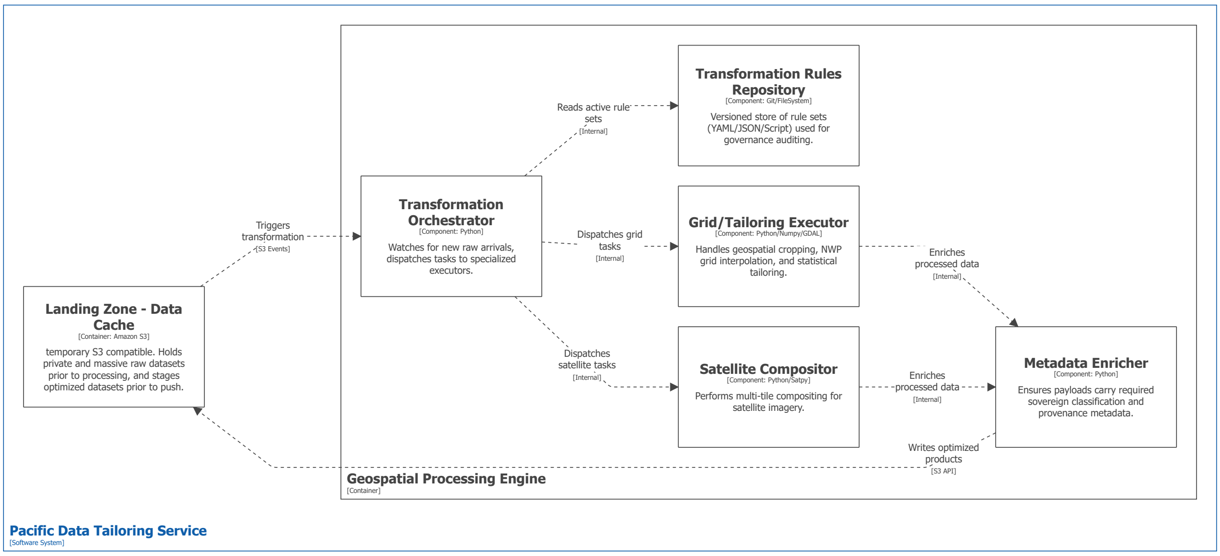
- landing zone for Global datasets that require tailoring
- landing zone for PICT data sets destined for the IFP and are not to be shared via WIS2.
- Holds raw datasets and transformed data sets
- S3-compatible backend (RustFS or AWS S3)
- Partitioned using the WIS2 topic hierarchy and data classification, the Data Lake can be partitioned to organise data by source, type, and classification.
  e.g.
- `pdts_cache_bucket/{pict-data-centre-id}/data/recommended/weather/surface-based-observations/2026-06-02-2212-{private station code}-AUTO-minute-swob.xml`
- `pdts_cache_bucket/uk-metoffice-wmc/data/core/weather/prekdiction/forecast/medium-range/deterministic/global/wind-speed-gust-max-3h/forecast/10/T+39/20260609T000000Z.global.forecast.-50_-90_70_0.2026-06-10T15_00_00Z.wind-speed-gust-max-3h.10.grib2/20260609T040147Z.8548`
- Provides object event notifications which will be used to trigge downstream processing by other PDTS components
- Automatic cleanup (raw data retained 24-48 hours only)

Benefits:

- Decouples fetching from processing (if processing delayed, fetching continues)
- Event-driven triggers (scalable, efficient)

- Enables parallel processing (multiple sources simultaneously)

3.3.3 Geospatial Processing Engine¶



Component View: Pacific Data Tailoring Service - Geospatial Processing Engine
Component-level view of the Geospatial Processing Engine.
Tuesday, 9 June 2026 at 6:46 pm New Zealand Standard Time

Responsibility: Transform massive global datasets into Pacific-optimized subsets.

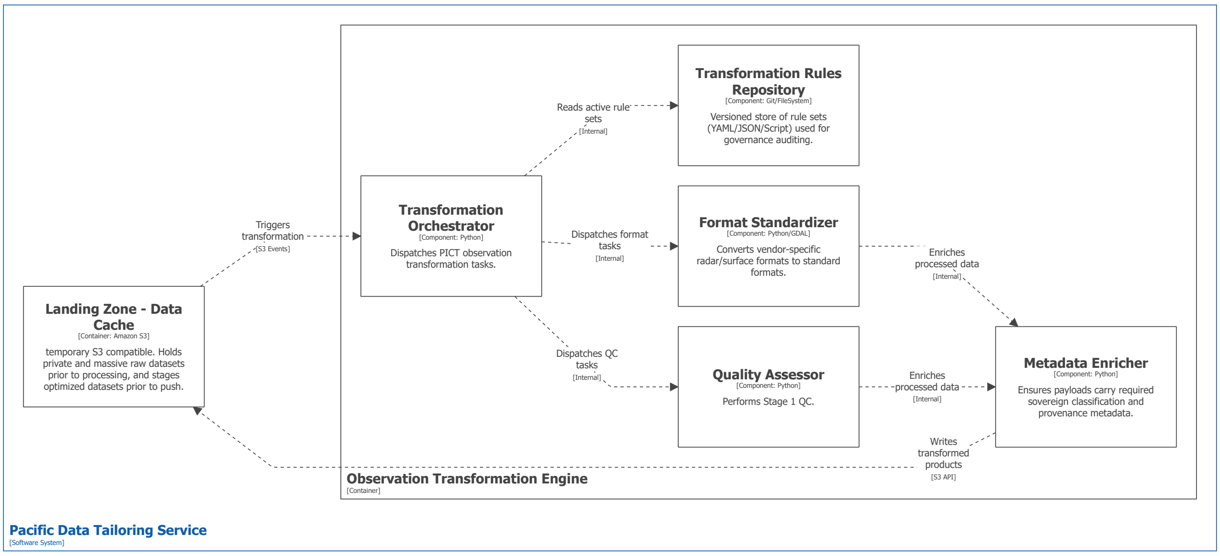
Key Transformations:¶

- NWP Grid Cropping
- Satellite Compositing
- Parameter Extraction
- Extract only forecast parameters relevant to Pacific (wind, pressure, temp, precip)
- Omit irrelevant parameters (reduces dataset further)
- Vertical interpolation if unavailable in Visual Weather (model levels → pressure levels for compatibility)
- Compression & Format
- NetCDF4 with zlib compression
- GeoTIFF for satellite (COG - Cloud Optimized GeoTIFF for S3 efficiency)

Performance:

- Parallel processing (multiple sources simultaneously)

3.3.4 Transformation Engine¶



Component View: Pacific Data Tailoring Service - Observation Transformation Engine
Component-level view of the Observation Transformation Engine.
Tuesday, 9 June 2026 at 6:46 pm New Zealand Standard Time

Responsibility: Standardize PICT observations and prepare for publication.

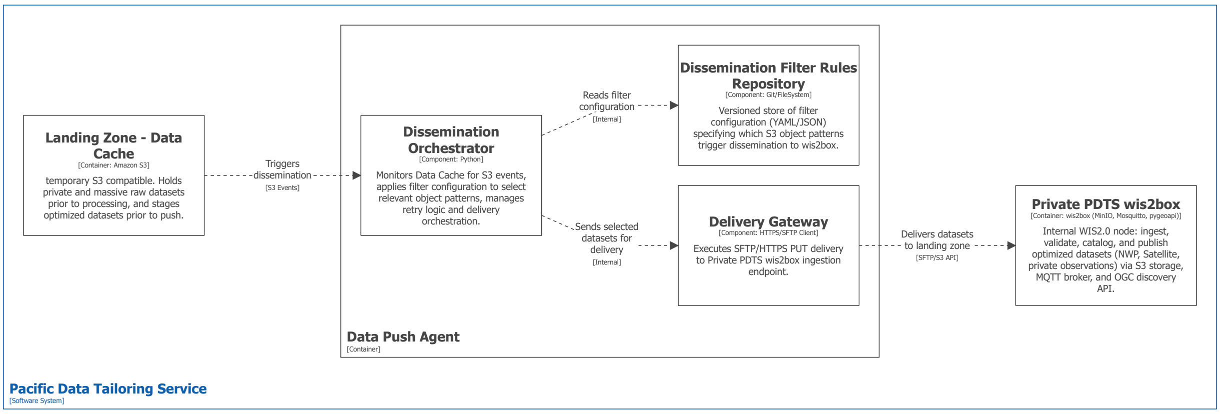
Transformations:

- QC and validation (remove obviously bad records)
- Format conversion (e.g. proprietary radar format → BUFR or NetCDF)

Technologies:

- Custom parsers for PICT radar formats
- Standard meteorological libraries (pygrib, netCDF4)
- Metadata schemas (CF conventions, WMO standards)

3.3.5 Data Push Agent¶



Component View: Pacific Data Tailoring Service - Data Push Agent
Component-level view of the Data Push Agent.
Tuesday, 9 June 2026 at 6:33 pm New Zealand Standard Time

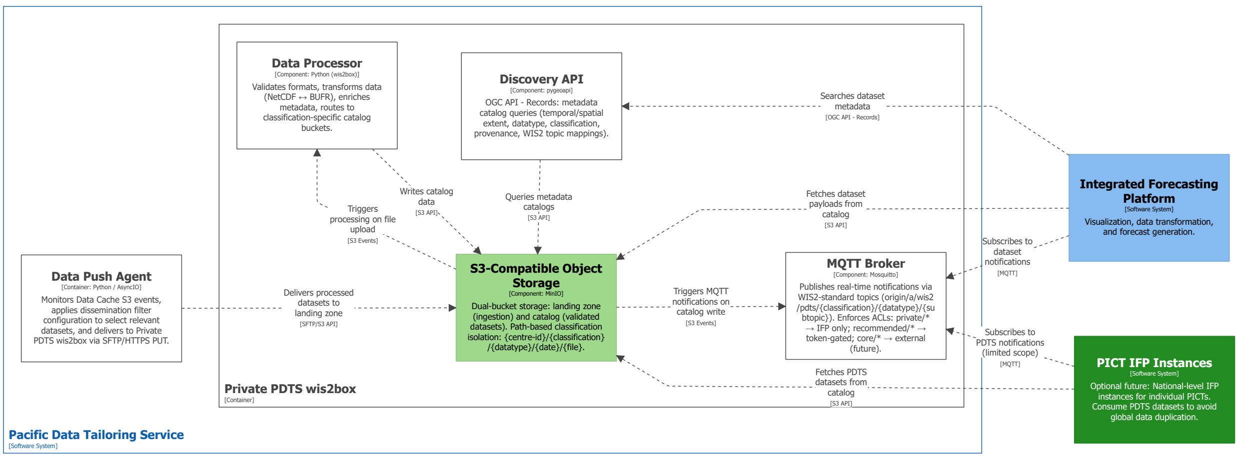
Responsibility: Monitor Data Cache and push optimised and select data sets to Private PDTS wis2box.

Workflow:

1. **Monitor:** S3 events on new files in optimized zone
2. **Filter:** Apply dissemination rules (what gets published when)
3. Copy or move object to PDTS WIS2 box ingest bucket

- `pdts_wisbox-private-ingest_bucket/pdts-{pict-data-centre-id}/data/recommended/weather/surface-based-observations/2026-06-02-2212}-{private station code}-AUTO-minute-swob.xml`
- `pdts_wisbox-private-ingest_bucket/ppts-uk-metoffice-wmc/data/core/weather/prekdiction/forecast/medium-range/deterministic/global/wind-speed-gust-max-3h/forecast/10/T+39/20260609T000000Z.global.forecast.-50_-90_70_0.2026-06-10T15_00_00Z.wind-speed-gust-max-3h.10.grib2/20260609T040147Z.8548`

3.3.6 Private PDTS wis2box¶



Component View: Pacific Data Tailoring Service - Private PDTS wis2box
Component-level view of the Private PDTS wis2box (Level 3 C4 Model).
Tuesday, 9 June 2026 at 6:33 pm New Zealand Standard Time

Responsibility: WIS2-compliant broker for IFP-only data.

Characteristics:

- **Internal only:** No external connections or subscriptions; separate from public WIS2Node MQTT broker
- **MQTT Broker:** Publish/subscribe model (real-time notifications) with username/password authentication

- **HTTPS Cache:** Bulk data pulls (for IFP's cache pulls)
- **Topic Structure:** WIS2-compatible topic hierarchy
- **Authentication:** Username/password credentials restrict subscriptions to authorized parties only

Important Distinction: This is a **separate, private MQTT broker** from the public WIS2Node/Global Broker. Unlike the public brokers (which use `everybody:everybody` placeholder credentials), this private broker uses **actual username/password authentication** to restrict access to authorized consumers only.

Topic Examples:
origin/a/wis2/{country-node}/data/recommended/weather#

```
origin/a/wis2/pdts-{pict-data-centre-id}/data/recommended/weather/surface-based-observatio
```

Access Control (PDTS Private WIS2Box):

- **Separate MQTT Broker:** Different from public WIS2Node broker
- **IFP Authorisation:** IFP has super-consumer authorisation → Can subscribe to ANY private topic
- **No External Access:** No other system can access these topics (username/password authentication enforces)
- **External Subscription Blocked:** External subscriptions rejected without valid credentials

Why WIS2-Compliant (But Private)?

- Enables future federation (if other systems granted super-consumer status)
- Maintains consistent interfaces (same MQTT/HTTPS protocol as Global WIS2)
- **Note:** This is a **different authentication model** than the public WIS2Node broker
- Simplifies IFP implementation (same ingestion code for both Global + Private)

PDTS Operational Context¶

Deployment: AWS ECS + Fargate (serverless, auto-scaling)

Scaling:

- Compute scales based on data volume and processing time
- Storage scales with global dataset retention

- No fixed hardware footprint (serverless cost model)

Operated By: SPREP (with regional technical team support)

Network: AWS region (currently `ap-southeast-2` Sydney, or other regional choice)

Appendix: Terminology & Cost Categories Reference¶

Terminology Glossary¶

Term	Definition	Context	Owner
NMHS (National Meteorological and Hydrological Service)	International standard term for the national-level meteorological and hydrological authority in any country. In the WRP/IFP context, refers to the meteorological service operating in any individual PICT country.	One NMHS per PICT country (e.g., Fiji Meteorological Service, Samoa Observatory). Operators of national forecasting infrastructure and data collection systems.	Architecture / Governance
National NMHS Forecaster	A forecaster employed by a national meteorological service in a specific PICT country. Operates at country-level scale; uses remote/web-based tools for forecasting and product generation.	Each PICT country has multiple National NMHS Forecasters across different disciplines (aviation, marine, hydrology, general). All National NMHS Forecasters collectively use the Online Weather SPA (web-based frontend).	Operations / Training
RSMC Forecaster	A forecaster employed by the Regional Specialized Meteorological Centre (RSMC) based in Fiji. Operates at regional Pacific	Fiji Meteorological Service forecasting team (typically 2-4 senior forecasters). Use Visual Weather	RSMC Fiji

Term	Definition	Context	Owner
	scale; has leadership/oversight authority over national forecasters; uses high-performance desktop tools for regional analysis and product generation.	Fat Client for intensive analysis and regional forecast generation.	

Cost Categories Glossary¶

This glossary is provided for quick reference during cost estimation and planning.

Capital Costs¶

Category	Definition	Examples	Typical Owner
IBL Licensing (Capital)	Initial purchase/acquisition of Visual Weather and/or Online Weather software licenses	License agreements, initial seat purchases	Procurement
Engineering (Initial Development)	Contractor and staff effort to design, build, and integrate custom platforms and services	Python orchestration, transform services, product dissemination orchestrator, PDTS fetch coordinator	Project Management / Contractors
Engineering (Initial Configuration)	Setting up initial operational configurations, workflows, data mappings, integrations	SOP workflows, PICT routing rules, dissemination adapters, WIS2Box configs, PDTS transformation rules	Operations / Architecture
Infrastructure Setup (Fiji)	One-time setup of on-premises infrastructure	Server procurement, RustFS deployment, networking,	Fiji IT / Infrastructure

Category	Definition	Examples	Typical Owner
		containerization foundation	
Infrastructure Setup (AWS)	One-time provisioning of AWS cloud infrastructure	VPC, security groups, IAM roles, S3 buckets, ECS cluster setup	AWS / Infrastructure
Training & Change Management (Initial)	Initial training programs and change management communications	Forecaster training, stewards certification, runbooks, SOPs	Operations / Training

Operating Costs¶

Category	Definition	Examples	Typical Owner
IBL Licensing (Annual)	Annual maintenance, support, and license renewal for Visual Weather and/or Online Weather	Maintenance contracts, vendor support SLA, license renewal fees	Procurement
AWS Infrastructure	Recurring compute, networking, and managed service costs on AWS	ECS/Fargate compute, data transfer, CloudWatch, managed service overhead	Infrastructure / Finance
AWS Storage	Recurring S3 storage costs for all data tiers and retention	Active 30-day S3, 1-year archive, tiering, backups, data transfer	Infrastructure / Finance
Fiji On-Premises	Recurring costs for on-premises infrastructure operations	Server maintenance, OS patching, power, internet connectivity, local IT support	Fiji IT / Infrastructure
Operational Stewardship	Ongoing staff FTE for operations, governance, and	WIS2 stewards, PDTS operations, IFP AWS operations, incident response,	Operations / RSMC Fiji

Category	Definition	Examples	Typical Owner
	support of all three systems	configuration evolution	
Git/DevOps	Recurring version control and CI/CD infrastructure costs	GitHub/GitLab subscriptions, CI/CD runners, automation tools	Infrastructure / DevOps
Monitoring/Alerting	Recurring monitoring and alerting infrastructure costs	CloudWatch, SaaS monitoring tools, alerting systems	Infrastructure / Operations