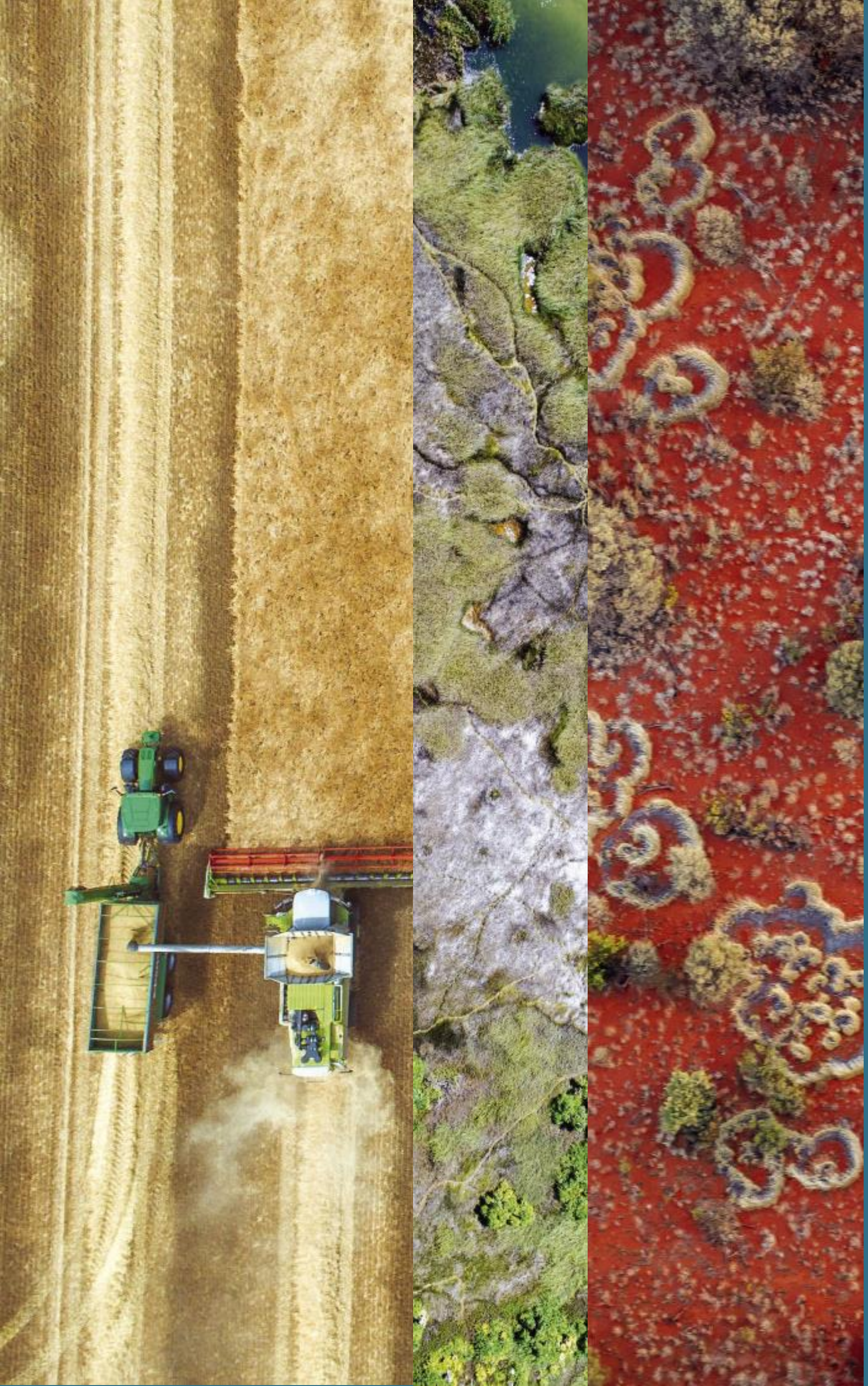


Australian Government
Geoscience Australia

Ginan Workshop

IGNSS 2024 - February 6th, 2024

Ginan – a precise point positioning software toolkit and position correction service

Guides for the Workshop

- Please find all guides written up for this workshop at the following link:

- https://linktr.ee/ginan_GA

- Click on the first link to be directed to a Dropbox Folder with Word Documents outlining:

    - Setting Up Your Workspace
    - Running Ginan
    - Setting Up Docker Desktop (For those using Docker)

NPIC - Ginan Workshop Agenda

Intro ~15 min (Simon)
- What is Ginan
- Major functionality

Ginan Installation ~ 45 min (Ron)
- Supported platforms
  - Linux, Mac, Rpi, Windows (via WSL)
- Docker, Native

Break 15 min

Configuring and Running Ginan ~ 45 min (Ron/Simon)

Hands on Ginan ~ 60 min (Ron/Simon/Jami…)

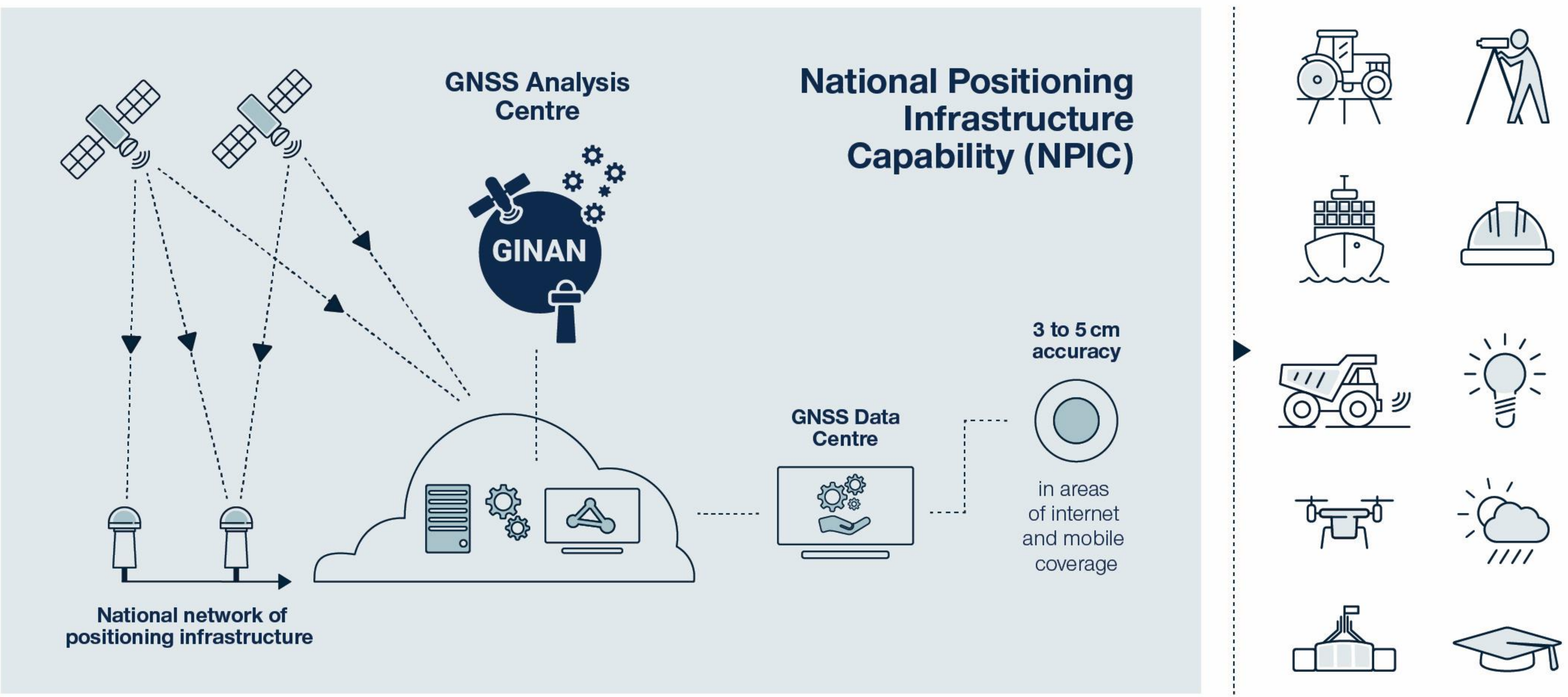
The Ginan name

- Ginan comes from the Wardaman people of Northern Territory
- Is a Wardaman word for a red dilly-bag filled with songs of knowledge
- Is the fifth-brightest star in the Southern Cross
- The Southern Cross helped the First Australians to navigate

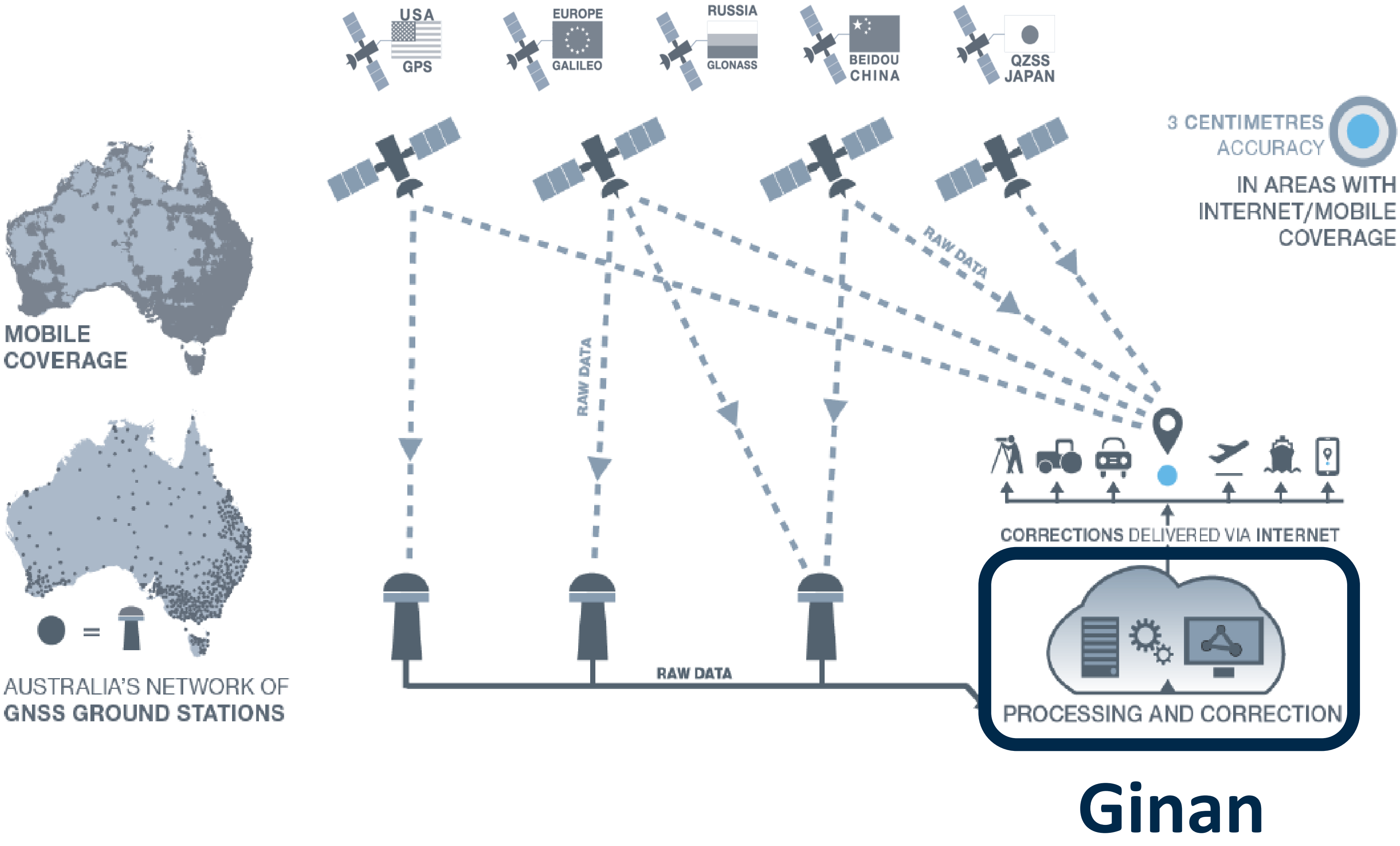
What is Ginan?

Positioning Australia (PA)
National Positioning Infrastructure Capability (NPIC)

“Accurate and reliable positioning for everyone”



The Ginan Concept



Ginan - Geoscience Australia's (GA) GNSS Analysis Centre Software

- Part of GA's Positioning Australia (PA) National Positioning Infrastructure Capability (NPIC)
- Open-source software toolkit for precise positioning and navigation
- Multi-GNSS data processing and analysis capability
- Undifferenced, State Space Representation (SSR) using Precise Point Positioning (PPP) methodology
- Capable of delivering precise positioning products and services for post processed and real-time applications
- Enables centimetre level accuracy positioning in areas with mobile phone/internet coverage

Methodology - PPP (SSR) vs RTK (OSR)

$$P^{s}_{r,f} = \rho^{s}_{r} + c\left(\cancel{\delta\tau^{q}_{r}} - \cancel{\delta\tau^{s}}\right) + \cancel{T^{s}_{r}} + \mu_f I^{s}_{r} + \cancel{d^{q}_{r,f}} + \cancel{d^{s}_{f}} + \cancel{O^{s}} + \varepsilon^{s}_{r,f}$$

- P - Observed range between satellite & receiver
- ρ - Geometric distance between satellite and receiver *
- c - Speed of light
- $\delta\tau$ - Receiver and satellite clock offsets
- T - Troposphere delay between satellite and receiver
- μ - Frequency dependent ionosphere delay factor
- I - Ionosphere delay between satellite and receiver
- d - Receiver and satellite hardware signal biases
- O - Satellite orbit errors
- ε - Range measurement noise

s, r, f, q – satellite, receiver, frequency, constellation indices

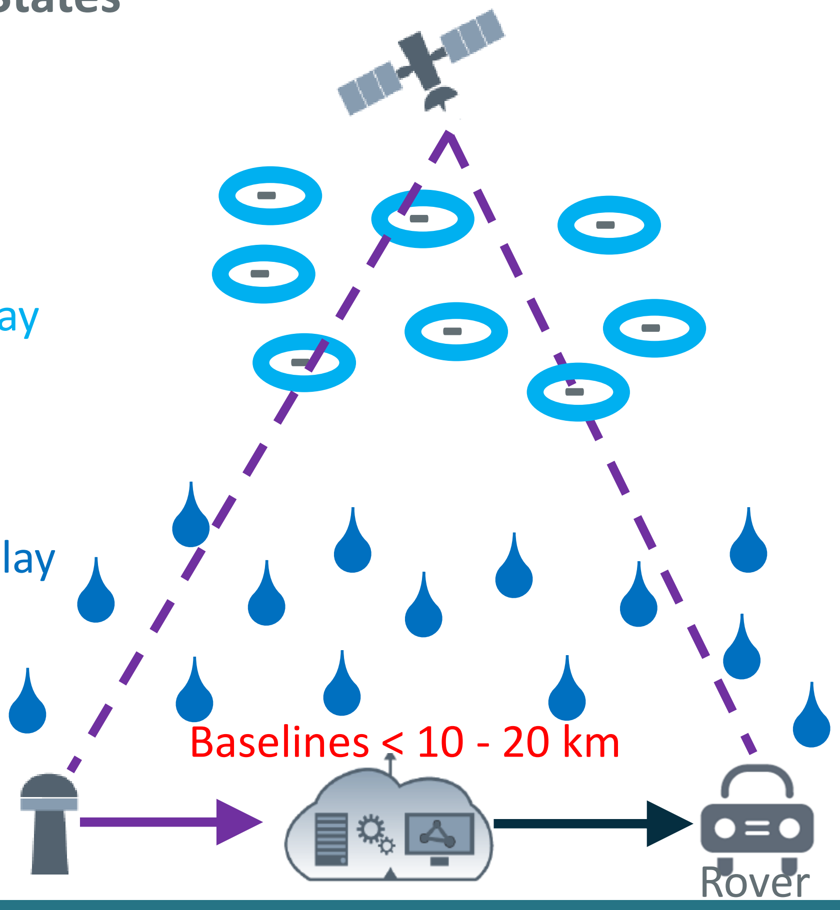
* The geometric range ρ is what is used for positioning
More precisely known ρ's == more precise positioning

Transmitted

Computed

GNSS model States

- Sat signal bias
- Sat clock error
- Sat orbit error
- Ionosphere delay
- Troposphere delay
- Rec signal bias
- Rec clock error



OSR vs SSR positioning

Observation Space Representation (OSR) - RTK

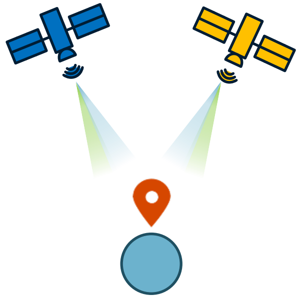
- 👍 Centimetre level accuracy
- 👎 Dense base station network
- 👎 Local coverage
- 👎 High bandwidth
- 👎 Two-way communication
- 👎 Not scalable
- 👍 Fast convergence

State Space Representation (SSR) - PPP

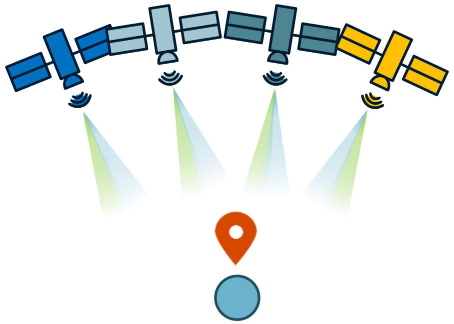
- 👍 Centimetre level accuracy
- 👍 Sparce base station network
- 👍 Global coverage
- 👍 Low bandwidth
- 👍 One-way communication
- 👍 Easily scalable
- 👎 Slow convergence

Key deliverables and milestones

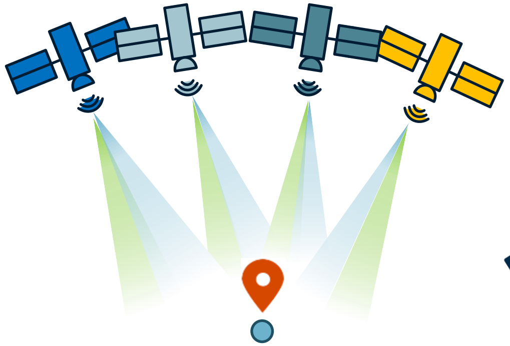
- **Ginan v1:** Dual frequency, multi-GNSS constellations, undifferenced, ionosphere-free, ambiguity resolved, real time configuration.
    - Operational release **July 2022**



- **Ginan v2:** Undifferenced, uncombined, ambiguity resolved, real-time configuration in network and user platforms, integrated combination platform.
    - Operational release by **May 2023**



- **Ginan v3:** Integration of Ionosphere & Troposphere models to achieve precise positioning in a shorter timeframe i.e. quick convergence time.
    - Operational release **December 2023**



Australian Government
Geoscience Australia

Ginan Components

Ginan

- Configuration
- Data Input
- Modelling + Prediction
- Filtering + Estimation
- Error Handling
- Augmentation
- Products Output

- Scripts + Visualisation

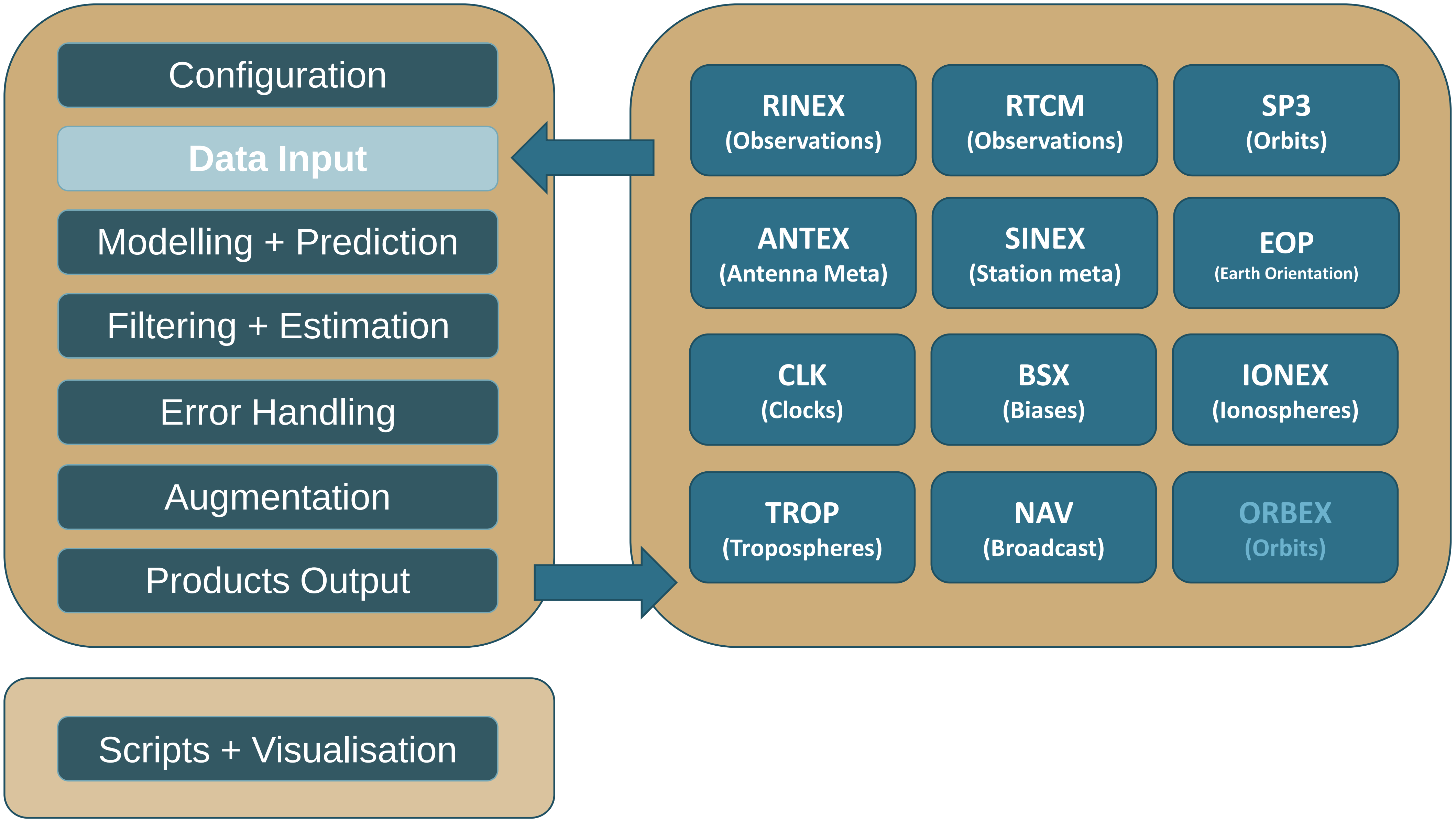
Ginan

- Configuration
- Data Input
- Modelling + Prediction
- Filtering + Estimation
- Error Handling
- Augmentation
- Products Output

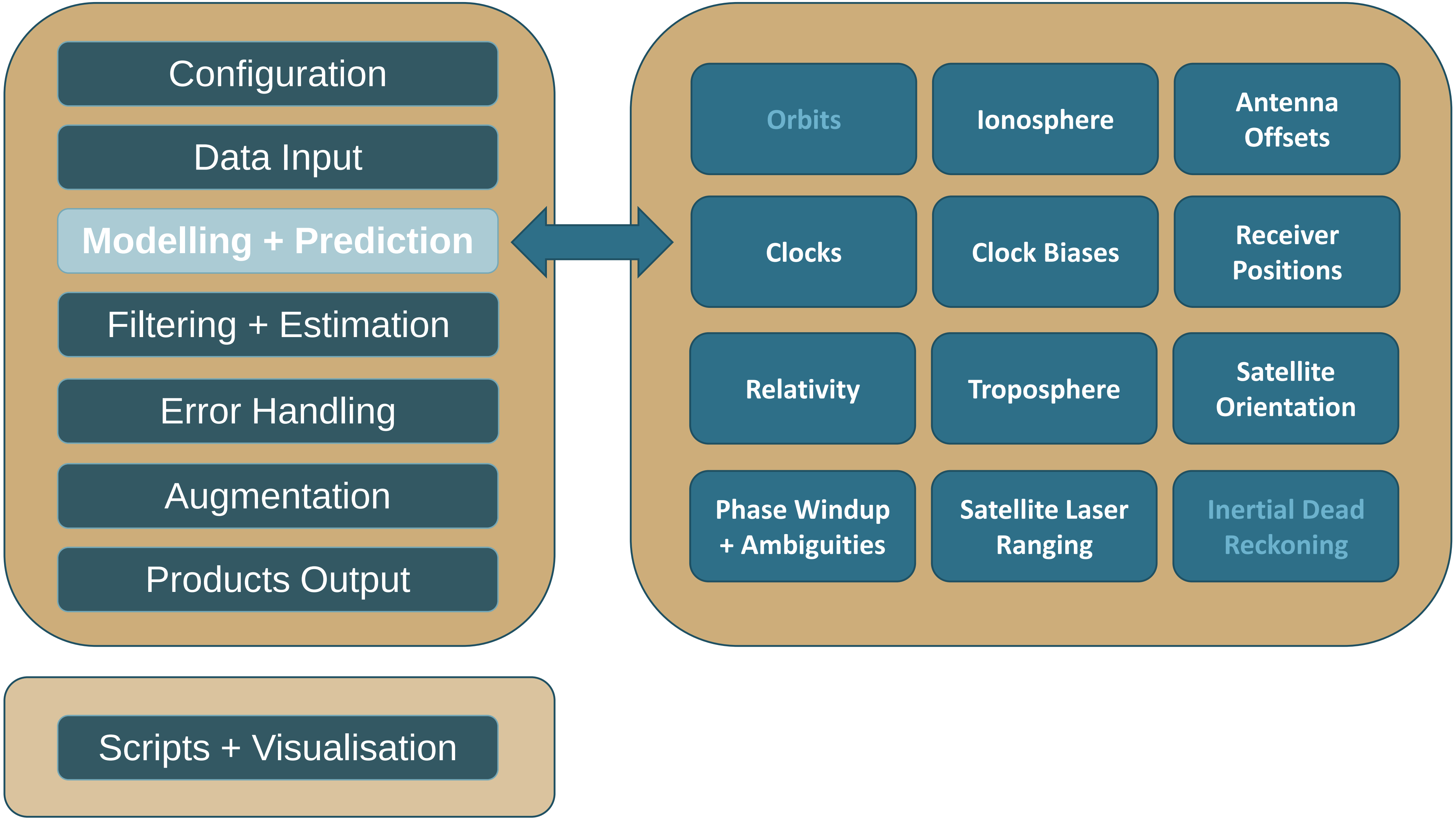
- Scripts + Visualisation


```yaml
#YAML configuration file

input_files:

    root_input_directory: products/

    atxfiles: [ igs14_2045_plus.atx ]
    snxfiles: [ "igs*.snx"          ]
    blqfiles: [ OLOAD_GO.BLQ        ]
    sp3files: [ igs20624.sp3        ]
    clkfiles: [ igs20624.clk_30s    ]
    bsxfiles: [ TUG0R03FIN_20191990000_01D_01D_OSB.BIA ]

station_data:

    root_stations_directory: data/

    rnxfiles:
        - "ALIC*.rnx"
        - "AGGO*.rnx"
        - "MAW*.rnx"

output_files:

    root_output_directory:          <CONFIG>/
```

Ginan

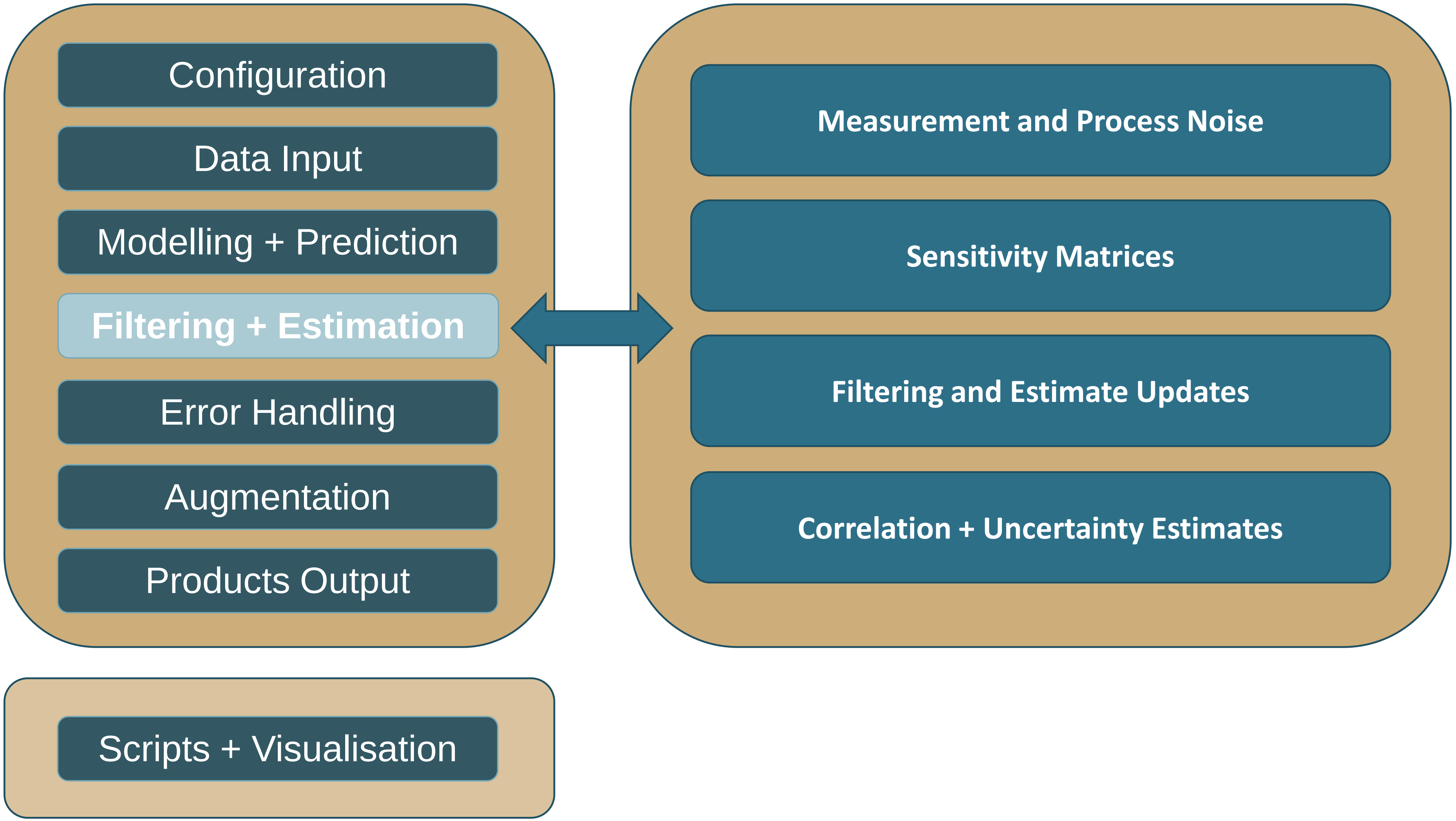

Configuration
Data Input
Modelling + Prediction
Filtering + Estimation
Error Handling
Augmentation
Products Output
RINEX
(Observations)
RTCM
(Observations)
SP3
(Orbits)
ANTEX
(Antenna Meta)
SINEX
(Station meta)
EOP
(Earth Orientation)
CLK
(Clocks)
BSX
(Biases)
IONEX
(Ionospheres)
TROP
(Tropospheres)
NAV
(Broadcast)
ORBEX
(Orbits)
Scripts + Visualisation

Ginan

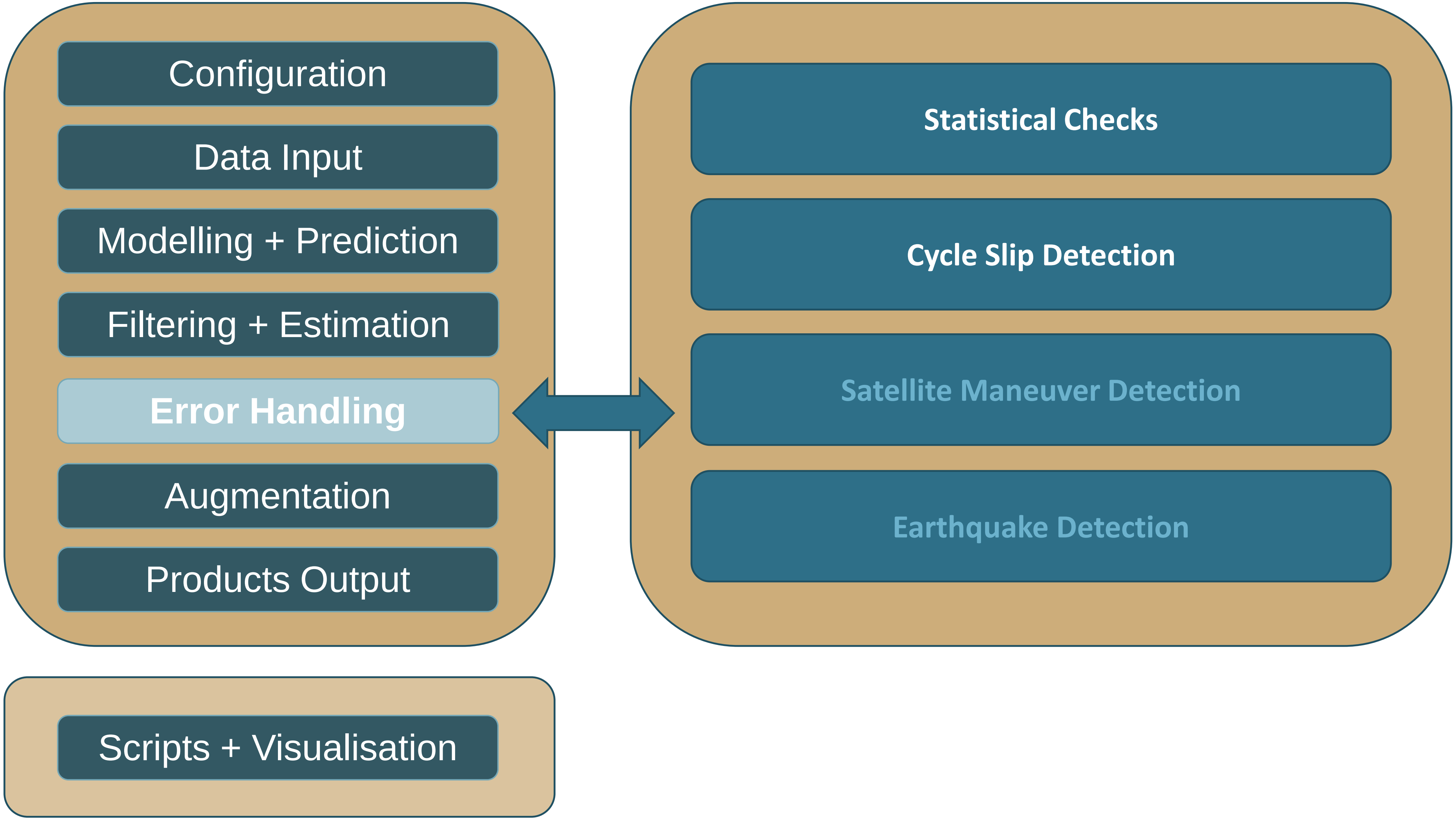


Configuration
Data Input
Modelling + Prediction
Filtering + Estimation
Error Handling
Augmentation
Products Output
Scripts + Visualisation
Orbits
Ionosphere
Antenna Offsets
Clocks
Clock Biases
Receiver Positions
Relativity
Troposphere
Satellite Orientation
Phase Windup + Ambiguities
Satellite Laser Ranging
Inertial Dead Reckoning

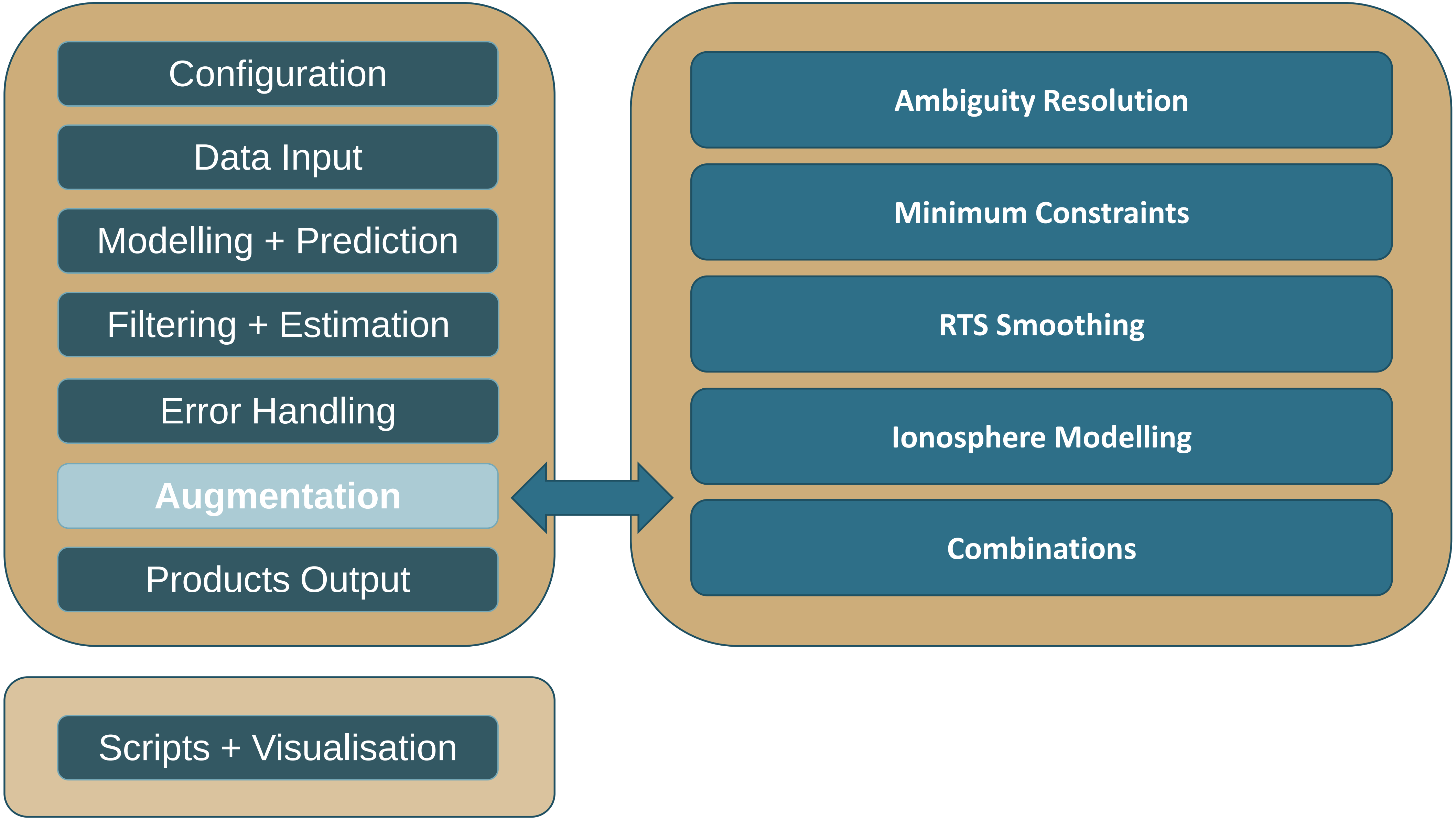
Ginan


Configuration
Data Input
Modelling + Prediction
Filtering + Estimation
Error Handling
Augmentation
Products Output
Scripts + Visualisation
Measurement and Process Noise
Sensitivity Matrices
Filtering and Estimate Updates
Correlation + Uncertainty Estimates

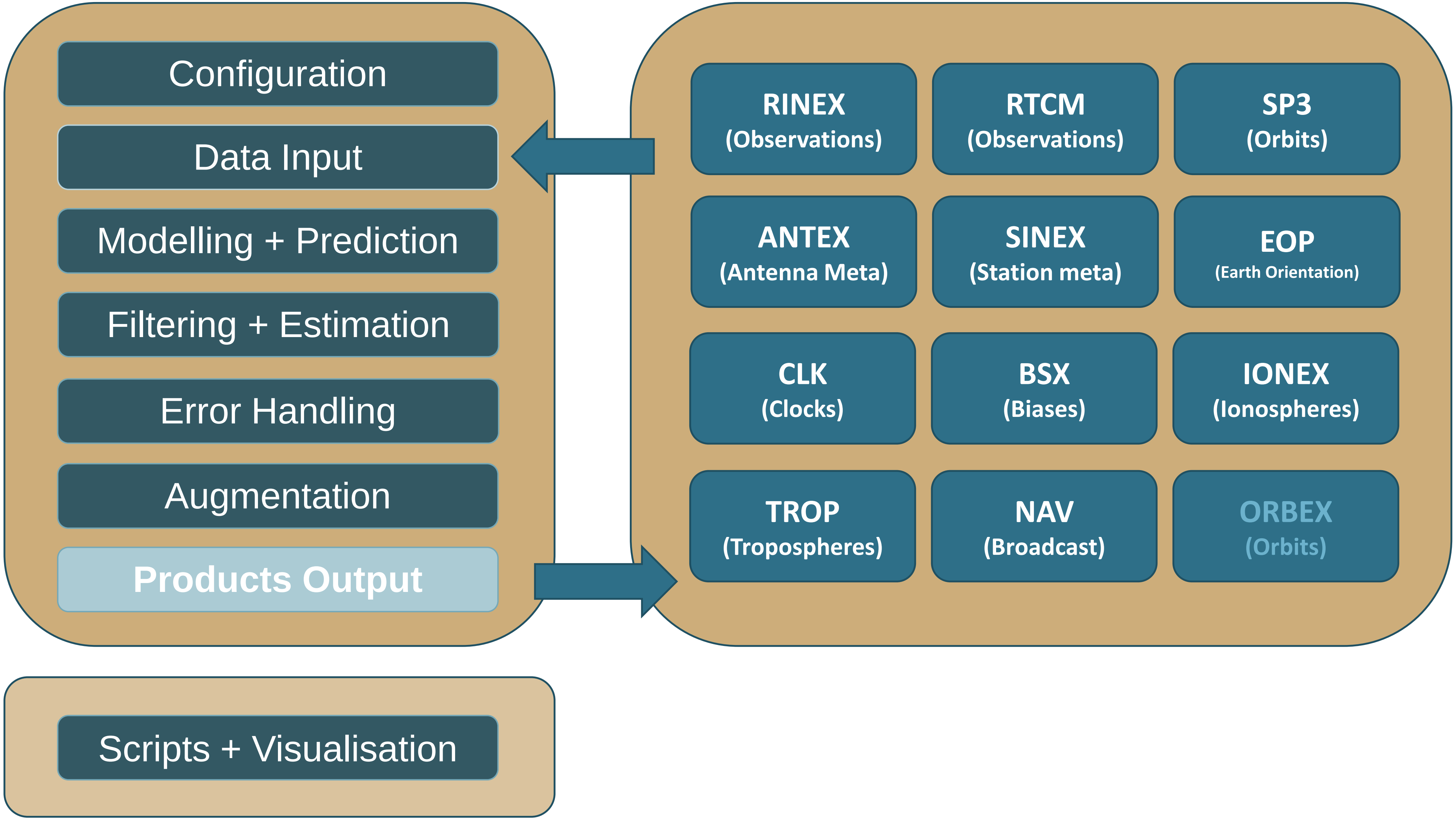
Ginan


Configuration
Data Input
Modelling + Prediction
Filtering + Estimation
Error Handling
Augmentation
Products Output
Scripts + Visualisation
Statistical Checks
Cycle Slip Detection
Satellite Maneuver Detection
Earthquake Detection

Ginan


Configuration
Data Input
Modelling + Prediction
Filtering + Estimation
Error Handling
Augmentation
Products Output
Scripts + Visualisation
Ambiguity Resolution
Minimum Constraints
RTS Smoothing
Ionosphere Modelling
Combinations

Ginan


Configuration
Data Input
Modelling + Prediction
Filtering + Estimation
Error Handling
Augmentation
Products Output
Scripts + Visualisation
RINEX (Observations)
RTCM (Observations)
SP3 (Orbits)
ANTEX (Antenna Meta)
SINEX (Station meta)
EOP (Earth Orientation)
CLK (Clocks)
BSX (Biases)
IONEX (Ionospheres)
TROP (Tropospheres)
NAV (Broadcast)
ORBEX (Orbits)

Australian Government
Geoscience Australia

Setting up Environment to Run Ginan

1. Prerequisites – Linux

- The workshop has been tested and written with Ubuntu in mind
- It has been tested on Ubuntu 22.04 and works for that version but should work for any Debian flavour
- Please let us know if you are having problems with your Linux distribution.

1. Prerequisites – Windows (using WSL)

1. Open Windows Powershell
2. To see if WSL is installed run the following command:

```
wsl --list --verbose
```

3. If not installed (or Ubunut not installed), go to (3a.) otherwise go to (4)
   a. Run the install command:

   ```
   wsl --install -d ubuntu
   ```

   b. Follow the on-screen instructions to set up your user

4. Open up WSL by running the command:

```
wsl
```

1. Prerequisites – MacOS

- The easiest way to Run Ginan on a Mac is to use Docker
- Therefore, the first step is to ensure that Docker is installed on your machine,
- Follow the instructions on the Docker website and the Docker Guide provided

2. Setting up Workspace – Create Directory

- Open up a Linux Terminal
- Choose a location to hold files used in this Workshop
- Create a new directory:

```
$ mkdir ginanworkshop
```

- Change directory:

```
$ cd ginanworkshop
```

2. Setting up Workspace – Update: Python, wget

- Update package list and install `wget` (if not yet installed)

```
$ sudo apt update
$ sudo apt install wget
```

- Upgrade python 3 and install virtual environment package (`python3-venv`)

```
$ sudo apt upgrade python3
$ sudo apt install python3-venv
```

2. Setting up Workspace – Create virtual python env

- Create a new virtual Python environment

```
$ python3 -m venv ginanenv
```

- Activate virtual environment

```
$ source ginanenv/bin/activate
```

- You should now see `(ginanenv)` on the terminal line

2. Setting up Workspace – Install python packages

- Ensure pip is running on the latest version:

```
$ pip install --upgrade pip
```

- Install the required python packages to run our scripts:

```
$ pip install click
$ pip install numpy
$ pip install requests
$ pip install Hatanaka
```

{Update: `pip install gnssanalysis` will be the only package necessary after 1 July 2024}

2. Setting up Workspace – Download Ginan Image

- Download the Ginan AppImage which can be used to run Ginan on the command line directly

```
$ wget https://github.com/GeoscienceAustralia/ginan/raw/develop-weekly-appimage/Ginan-x86_64.AppImage
```

- Make the file executable:

```
$ chmod 777 Ginan-x86_64.AppImage
```

2. Setting up Workspace – Download input files

- Download a configuration YAML file which tells Ginan what and how to run

```
$ wget https://raw.githubusercontent.com/GeoscienceAustralia/ginan/main/exampleConfigs/ppp_example.yaml
```

- Download the *auto_download_PPP.py* script and a dependency script which downloads input files based on datetime:

{**Update:** The dependency script `gn_functions.py` will not be necessary after 1 July 2024}

```
$ wget https://raw.githubusercontent.com/GeoscienceAustralia/ginan/main/scripts/auto_download_PPP.py
$ wget https://raw.githubusercontent.com/GeoscienceAustralia/ginan/main/scripts/gn_functions.py
```

2. Setting up Workspace – Test environment

- Test that the *auto_download_PPP.py* script runs correctly:

```
$ python3 auto_download_PPP.py --help
```

- Test that Ginan runs:

```
$ ./Ginan-x86_64.AppImage --help
```

Downloading Inputs and Editing the YAML

3. Downloading Reference Inputs and RINEX Data

- In this workshop we will use the *auto_download_PPP.py* script to download all the necessary model reference inputs as well as the RINEX data

- This script has a number of options for downloading specific input files, like model files, SP3, CLK, BIA files, etc.

- The observation data we will be using comes from one of the Continuously Operating Reference Stations (CORS) in the Geoscience Australia network–specifically the ALIC00AUS station in Alice Springs, NT.

3. Downloading Reference Inputs and RINEX Data

- In your terminal window, run the following command:

```
$ python3 auto_download_PPP.py --target-dir products --preset igs-station --station-list ALIC --start-datetime 2024-01-06_00:00:00 --end-datetime 2024-01-06_23:59:30 --product-dir products --rinex-data-dir data
```

- This tells the script to:
  - download files to the `/products` directory
  - using the preset `igs-station`
  - which allows us to specify an IGS station (here: `ALIC`)
  - for data between midnight 6 Jan 2024 and the end of the same day.

- It will also gather all model reference files and orbits, clocks, biases, SNX positions from CDDIS

4. Editing YAML – Input Files

- Next is to edit the YAML file. This is the configuration (or control) file for running Ginan

- We will begin by editing the `inputs:` section so that Ginan will grab the files we downloaded

- First, we will **comment out** the following:
  - the weekly SNX file under `snx_files`
  - the CLK, BSX and SP3 files under `satellite_data`
  - the RINEX files under `gnss_observations:rnx_inputs`

- Next, we will **uncomment** the following:
  - the **.SNX* wildcard under `snx_files`
  - the **.CLK*, **.BSX* and **.SP3* wildcards under `satellite_data` as well as uncomment `nav_files` and its ".rnx" wildcard from the `satellite_data` field
  - the **.rnx* wildcard for RINEX data under `gnss_observations:rnx_inputs`

4. Editing YAML – Outputs and Processing Options

- We will leave all `outputs:` options as they are (you can rename files here if you wish)
- The next section is `processing_options:`, and the primary thing we will do here is turn off processing the Galileo constellation
- We do this by setting `process:` to false, which is located under `processing_options:gnss_general:gal`
- This is in the interest of time for this workshop but for your own processing you can leave it on

4. Editing YAML – Outputs and Processing Options

- An important note here, especially when processing your own RINEX files, is the code_priorities: list for each constellation
  - You must ensure that the list is not too long (each code will be processed)
  - You must ensure that the codes you specify are in the RINEX data you are processing

- We will now check the *ALIC RINEX* file that we downloaded and adjust accordingly
- By default, the example YAML has the following set: `[L1W, L1C, L2W]`
- The L1W signal can be removed from our list

4. Editing YAML – Turn Off Mongo

- Lastly, we will turn off Mongo

- Scroll down to the end of the Document
- The last block is `mongo:`
- Here you will set the `enable:` field to `none`

- The Mongo Database is required if you use the Ginan – Exploratory Data Analysis (EDA) tool
- We will showcase this feature during the workshop but do not provide a walkthrough of getting it up and running
- Please ask us for further information later

Australian Government
Geoscience Australia

Running Ginan

5. Running Ginan

- With all data and YAML settings in place, we can run the following command to start Ginan processing:

```
$ ./Ginan-x86_64.AppImage --config ppp_example.yaml &>stdout.txt
```

- Above, we are using the Ginan AppImage to run Ginan
- Giving the option `--config` specifies where it should find configuration to run with
- At the end, we are piping what is output to screen to a file `stdout.txt` instead to browse later

- Once complete, we can browse through our new `/outputs` directory to see the results
- We will primarily concentrate on the GPX file that is output

6. Visualising Results

- We will use a simple online GPX viewer to see the results of our processing run:

https://www.maplorer.com/view_gpx.html

- This allows you to upload the GPX file from the /outputs directory and view it on a map
- To do this:
  - Click on "Choose File" and browse to the relevant location.
  - Choose the ".GPX_smoothed" file if possible (result of RTS smoothing)
  - Click the "View Profile" button at the bottom of the page
- A new window should appear with a map of the results

7. Use Your Own Data

- We have now covered the main instructions for this Workshop

- Hopefully, we have provided you with enough information for you to continue to process your own data

- Assuming we have time at the point in the Workshop, we can now come around and help you with your particular use case / RINEX file

8. Contribute to Ginan

- Ginan is an open-source piece of software: https://github.com/GeoscienceAustralia/ginan

- You can freely take it and use it for your own projects
- You can also contribute back to the GitHub Repository that we maintain
- Please consider making a Fork of the Codebase and contribute back via a Pull Request against our repository

- We have external contributions right now

- E.g. Andrew Cleland has built a python tool for automatically generating YAML configuration files and downloading required inputs based on a given RINEX file
- https://github.com/GeoscienceAustralia/ginan/pull/62

Australian Government
Geoscience Australia

Thanks for Attending the IGNSS 2024 Ginan Workshop!

Workshop Team:

Simon McClusky, Ronald Maj, Umma Zannat, Eldar Rubinov, Lachlan Ng, Thomas Maile

To contact us, please send an email to:

GNSSAnalysis@ga.gov.au