RadioSkySim: A Flexible and Scalable Python Framework for Radio-Astronomical Sky Simulation

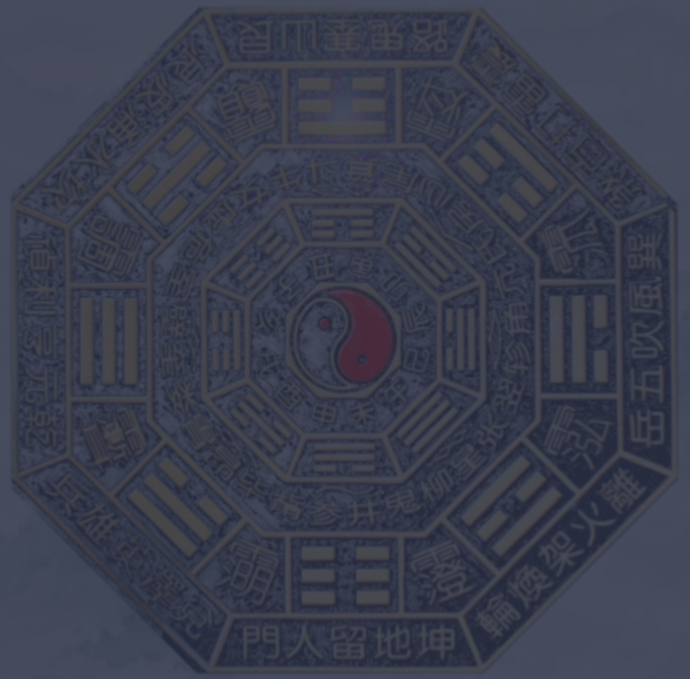


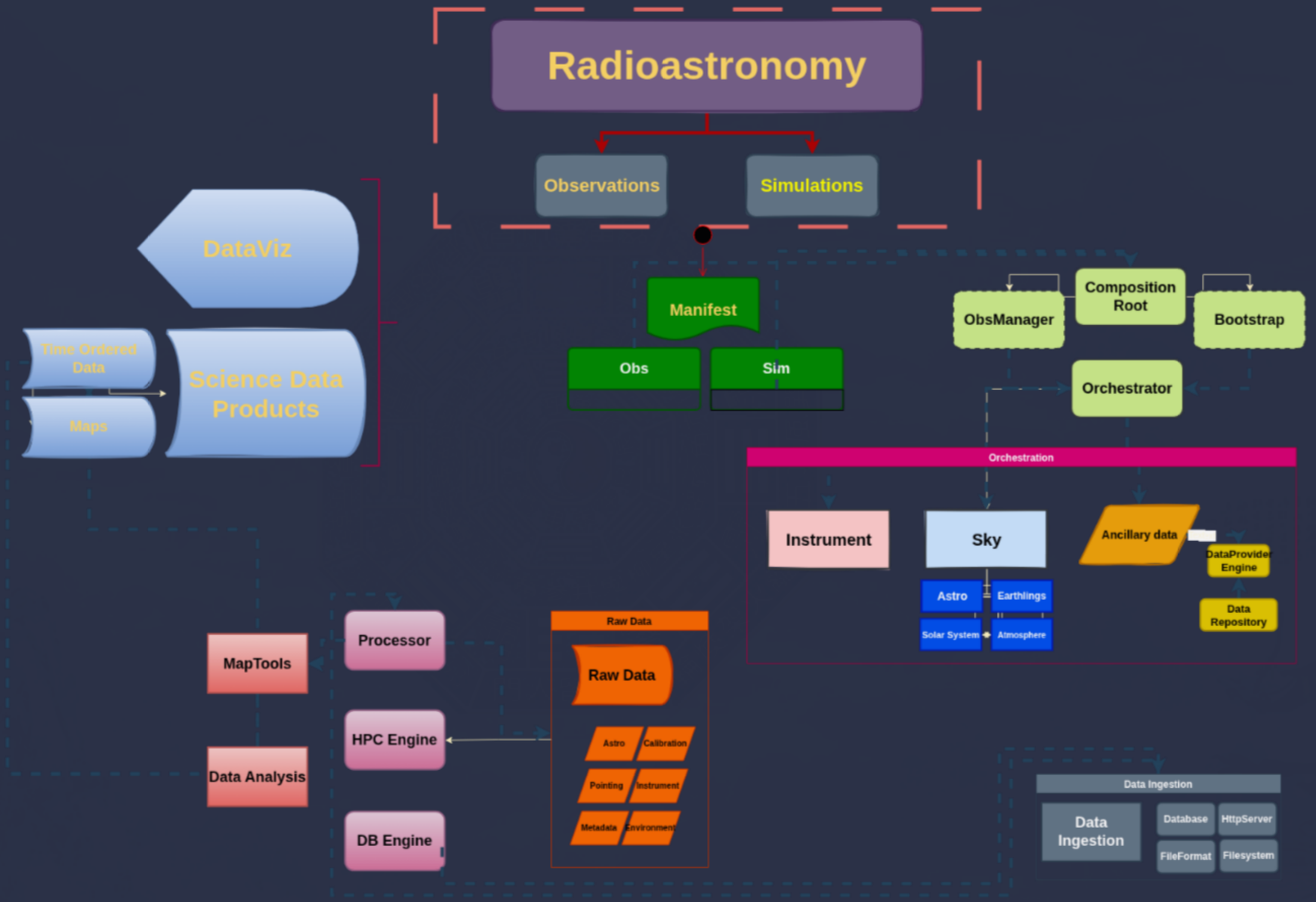
Luciano Barosi

Federal University of Campina Grande · BINGO Radio Telescope

lbarosi@df.ufcg.edu.br · @barosil

Radio Astronomy Simulations




Radioastronomy
Observations
Simulations
DataViz
Time Ordered Data
Maps
Science Data Products
Manifest
Obs
Sim
ObsManager
Composition Root
Bootstrap
Orchestrator
Orchestration
Instrument
Sky
Ancillary data
DataProvider Engine
Data Repository
Astro
Earthlings
Solar System
Atmosphere
MapTools
Data Analysis
Processor
HPC Engine
DB Engine
Raw Data
Raw Data
Astro
Calibration
Pointing
Instrument
Metadata
Environment
Data Ingestion
Data Ingestion
Database
HttpServer
FileFormat
Filesystem

Similar Proposals

Software	Sky/beam	TOD/noise	RFI	Maps/inference
HIDE & SEEK	●	●	●	●
limTOD	●	●	—	●
TOAST	◐	●	—	●
LiteBIRD	●	●	—	●
meer21cm	◐	◐	◐	●
Cosmoglobe	●	●	—	●

Radio Astronomy Simulations

Domain-Driven Design

Hexagonal Architecture

SOLID Principles

- Single Responsibility
- Open/Closed
- Liskov Substitution
- Interface Segregation
- Dependency Inversion

DRY — Don't Repeat Yourself



RadioSkySim

Simulação integrada para observações radioastronômicas.

Modularity

Componentes independentes

Scalability

Cresce com a demanda

Maintainability

Evolução previsível

Testability

Verificação automatizada

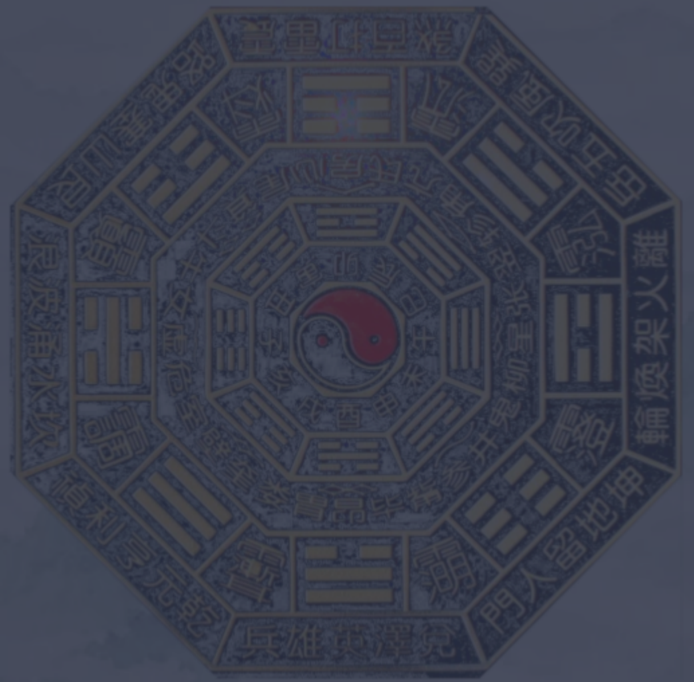
Tech Stack

Layer	Libraries	Comment
Runtime	Python 3.11–3.14, `uv` workspace	Good choice for a multi-package meta-repo; requires `uv sync`/CI discipline.
Domain/DTOs	`pydantic>=2`, `sqlmodel`, `SQLAlchemy>=2.0`	Consistent with `AGENTS.md` ("no `dataclass` in the domain").
Arrays/computation	`numpy`, `xarray`, `dask[distributed]`, `scipy`, `scikit-learn`	Correct stack for lazy arrays and single-node memory control.
Astronomy/cosmology	`astropy`, `skyfield`, `sgp4`, `healpy`, `pysm3`, `astroquery`, `pyvo`, `reproject`, `pyccl`, `camb`	Aligned with the domain (BINGO/HI/cosmology).
GNSS	`georinex`, `gnsstools4astro` (own package)	Covers RINEX NAV/OBS beyond the TLE catalog.
Persistence	PostgreSQL via `psycopg2-binary`/SQLModel; `h5py`/`h5netcdf`; `zarr`/`numcodecs`	Correct split: relational metadata vs. scientific arrays in HDF5/Zarr.
Visualization	`panel`, `holoviews`, `hvplot`, `datashader`, `jupyter-bokeh`, `matplotlib`, `cartopy`	Suitable for interactive exploration and notebooks.
Quality	`pytest`, `pytest-cov`, `ruff>=0.9` (`0.12.0` installed)	Lean setup today; see §7 for gaps.
Docs	`sphinx`, `myst-parser`, `myst-nb`, `pydata_sphinx_theme`	Complete Sphinx book in `docs/source/` (24 .md pages).

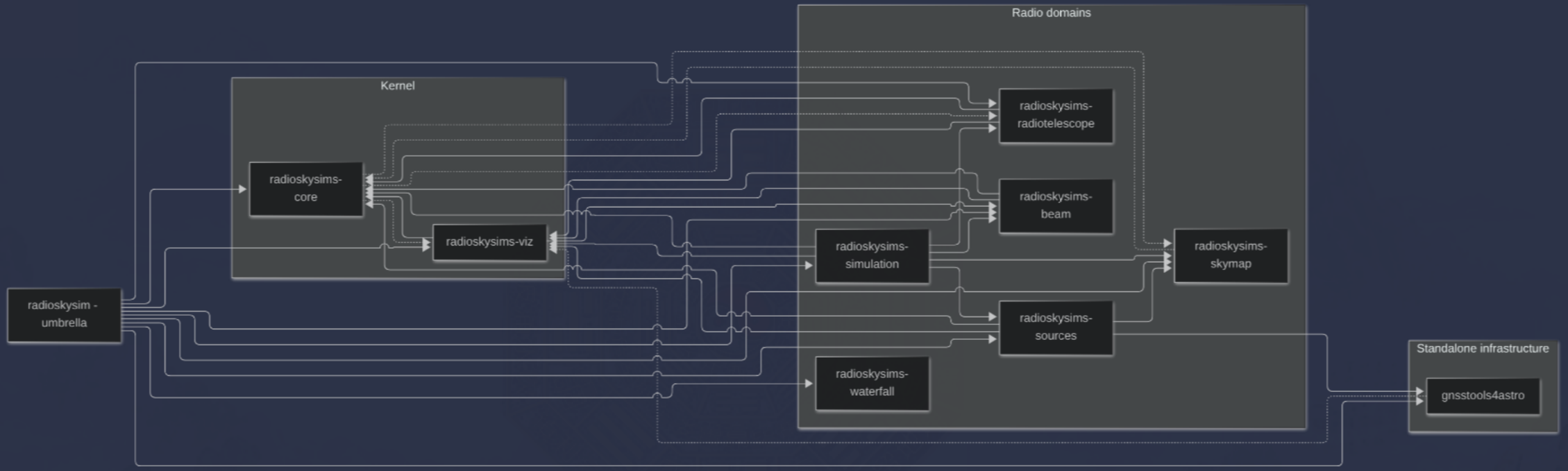
Bounded Contexts

Context	Responsibility	Main ports
Beam	Load/evaluate beam patterns	`BeamLoaderPort`, `BeamEvaluatorPort`
Satellite	Orbits (TLE), selection and tracks	`SatelliteRepositoryPort`, `GeometryProviderPort`
LocalSources	Local sources (Sun) and flux	`LocalSourcesPort`
RadioSources	Catalogs and spectral flux (TAP)	`RadioSourcesCatalogPort`, `RadioSourcesFluxPort`, `CatalogIngestPort`
TOD	Time Ordered Data synthesis	`TODSimulator`
SkyMap	HEALPix maps (generation/transformation/analysis)	`SkyMapGeneratorPort`, `SkyMapAnalyzerPort`, `SkyMapIOPort`
Simulation	End-to-end orchestration	aggregates the ports above; `SimulationRepositoryPort`
Waterfall	Waterfall tools (time × frequency)	`WaterfallLoaderPort`, `WaterfallStoragePort`

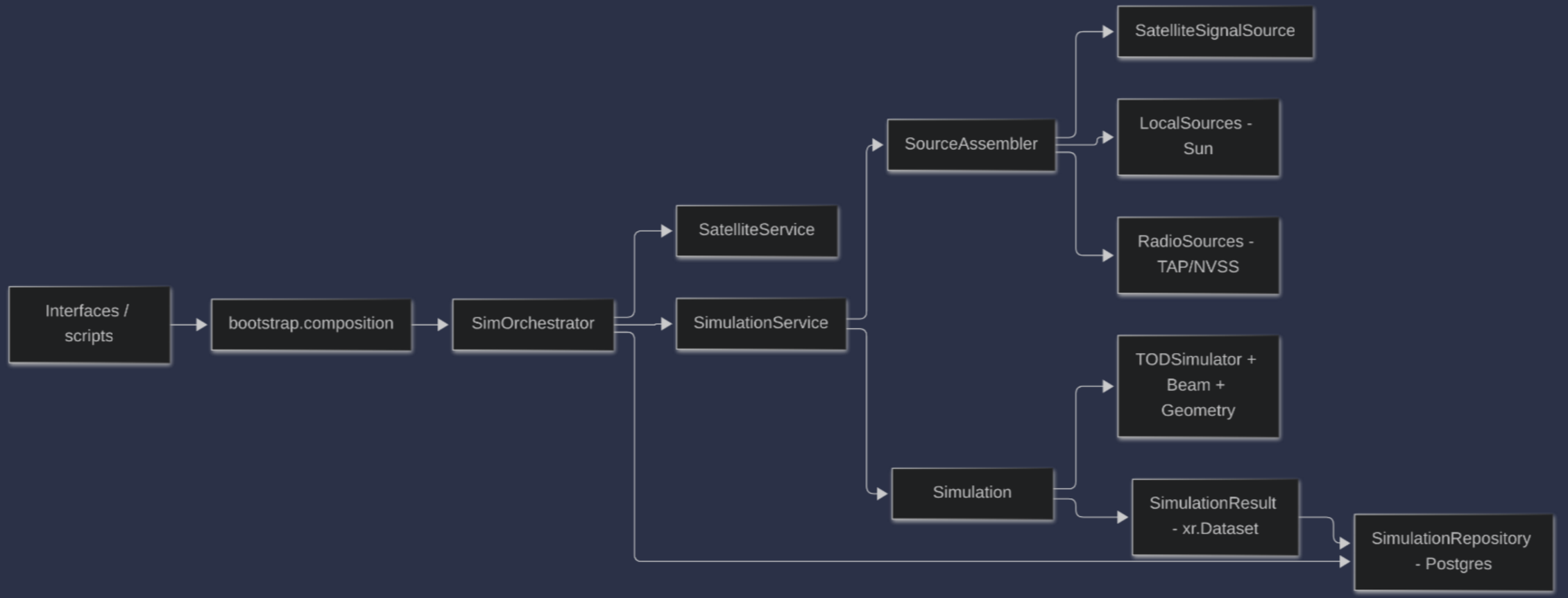
Architecture



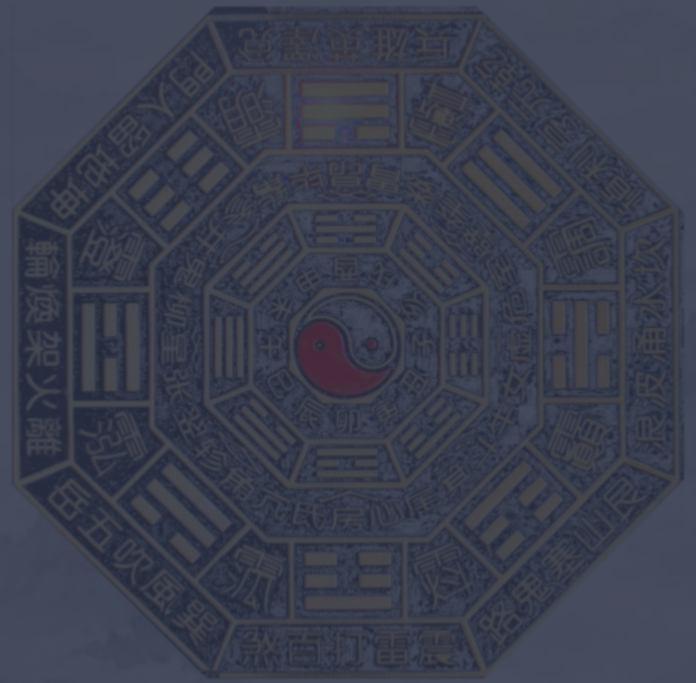
Ecosystem


Radio domains
Kernel
radioskysims-core
radioskysims-viz
radioskysims-radiotelescope
radioskysims-beam
radioskysims-simulation
radioskysims-skymap
radioskysims-sources
radioskysims-waterfall
radioskysim - umbrella
Standalone infrastructure
gnsstools4astro

Simulation Pipeline



Installation, Documentation and More



Project Structure

```
packages/
├── radioskysims-core/           # kernel: config, db, paths, common domain
├── radioskysims-beam/           # beam patterns and physical properties
├── radioskysims-radiotelescope/ # radio telescope profiles (antennas, pointing, backend)
├── radioskysims-sources/        # satellites, Sun, radio catalogs (TAP)
├── radioskysims-simulation/     # TOD + memory-aware processing
├── radioskysims-skymap/         # HEALPix maps
├── radioskysims-viz/            # visualization (matplotlib/holoviews)
├── radioskysims-waterfall/      # waterfall (channel × time)
└── gnsstools4astro/             # GNSS tools (own repo)

src/radioskysims/                # meta-package `radioskysim` (composition root)
├── bootstrap/                   # composition root + builders (DI)
├── interfaces/                  # CLI/GUI (skymap, tod, panel)
└── visualization/               # waterfall (legacy)
```

settings.toml

```toml
[paths]
raw_data_folder = "data/raw/"
auxiliary_data_folder = "data/aux/"
log_folder = "data/logs/"
skyfield_folder="data/aux/skyfield/"

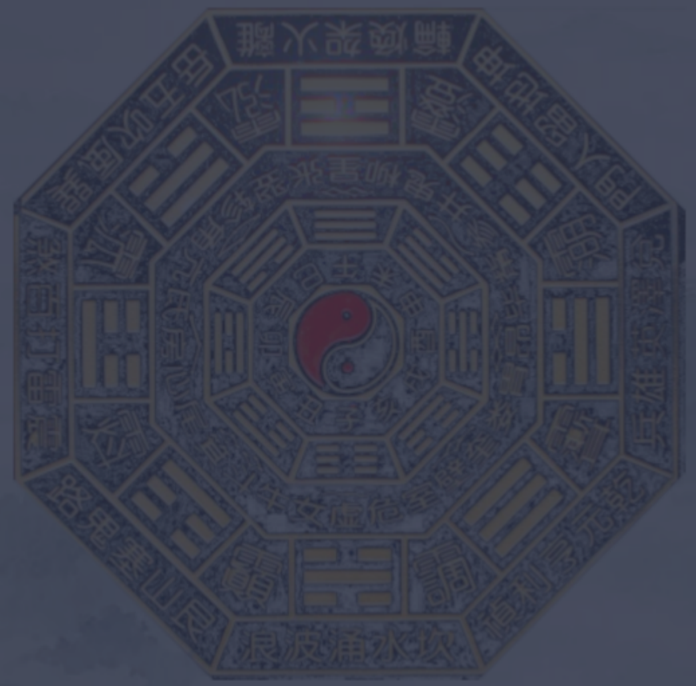
[ports]
db_port = 5432
panel_port = 8080

[radiotelescopes]
# ── DEFAULT profile (copy of uirapuru) ──────────────────────────────
# Profile used by CLI/notebooks when no --telescope is given
# (create_radiotelescope() and `radioskysims tod export` without --telescope).
[radiotelescopes.default]
name = "UIRAPURU Telescope"

[radiotelescopes.default.observer]
lat = -7.211637
lon = -35.90813
height = 530.0

[[radiotelescopes.default.antennas]]
name = "BINGO Horn"
# effective_area_m2 missing → A_e = λ²/Ω_A (Ω integrated from the beam); optional override.
frequency_range = [1000, 1250, 0.05]
```

Bounded Contexts and Domain



Radio Telescope



Solar Flux

RadioSources

GNSS Satellites

Metrics



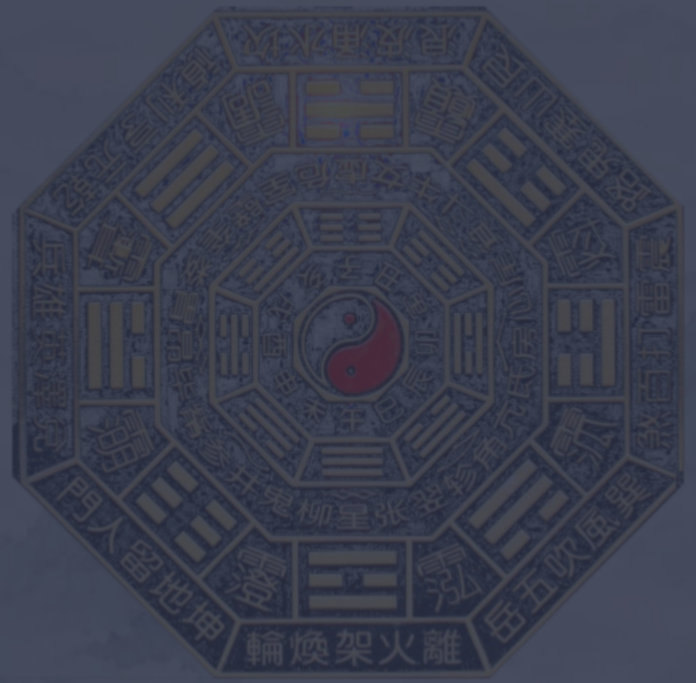
Breakdown by package (Sep/2026 + Submodules)

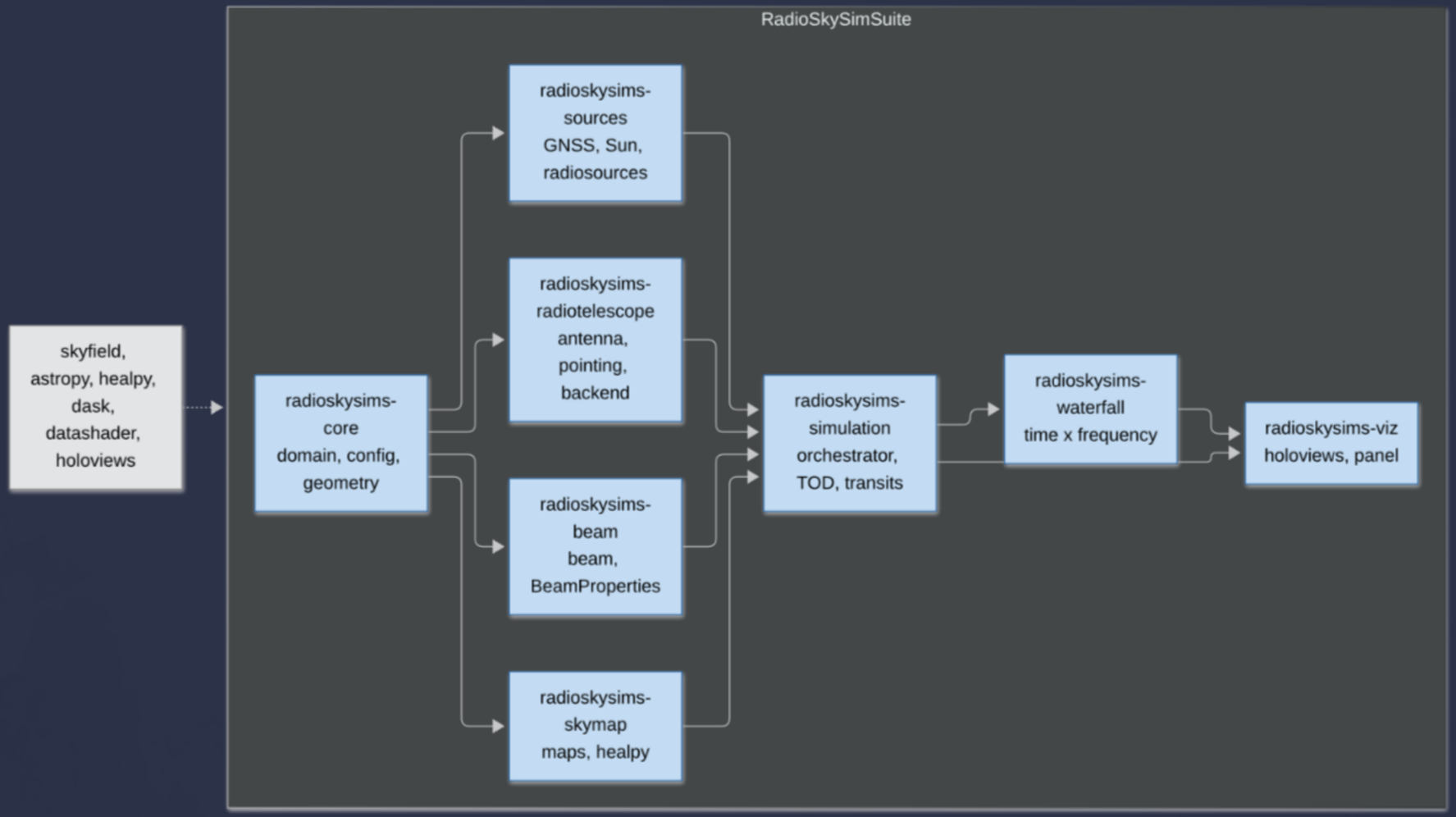
Package	Files	LOC	Functions	Type Coverage	Test Coverage
radioskysims (umbrella)	40	3.200	60+	~40%	~55%
radioskysims-core	35	5.800	150+	~60%	~70%
radioskysims-beam	25	4.500	120+	~55%	~60%
radioskysims-sources	30	6.200	140+	~50%	~65%
radioskysims-skymap	28	5.100	130+	~45%	~50%
radioskysims-radiotelescope	20	3.800	90+	~40%	~45%
radioskysims-simulation	45	12.500	200+	~35%	~60%
radioskysims-viz	32	6.400	100+	~50%	~55%
radioskysims-waterfall	28	7.200	150+	~40%	~65%
gnsstools4astro (submodule)	40	8.400	180+	~35%	~50%

Coverage by type

Category	Coverage (Sep)	Evolution	Status
Domain	~70%	Stable	✓ Good
Application	~65%	Stable	✓ Good

Running Simulations




RadioSkySimSuite
skyfield,
astropy, healpy,
dask,
datashader,
holoviews
radioskysims-
core
domain, config,
geometry
radioskysims-
sources
GNSS, Sun,
radiosources
radioskysims-
radiotelescope
antenna,
pointing,
backend
radioskysims-
beam
beam,
BeamProperties
radioskysims-
skymap
maps, healpy
radioskysims-
simulation
orchestrator,
TOD, transits
radioskysims-
waterfall
time x frequency
radioskysims-viz
holoviews, panel

```
[simulations]
# First-run window: 24 h, 120 s step (2881 samples at 30 s
# would produce ~113 GB of output; 120 s is the repo default for 1-day windows).
time_range = ["2026-10-11T00:00:00", "2026-10-12T00:00:00", 120]

[catalogs]
[catalogs.radiosources]
name = "Default Radio Sources Catalog"
url = "src/radioskysims/resources/catalogs/default_radiosources.csv"
[catalogs.spectral_indexes]
name = "Default Spectral Index Catalog"
url = "src/radioskysims/resources/catalogs/default_spectral_indexes.csv"

[radiosources]
query_str = "SELECT * FROM radiosources WHERE flux > 1"

[noisemodels]
pysm3_component = ["synchrotron"]

[satellites]
constellations = ["GPS", "GALILEO", "BEIDOU", "GLONASS"]
satellites = []

[signals]
# EIRP recomputed from BINGO_ASSEMBLY/notebooks/04_GNSS_SIGNAL_ESTIMATION.ipynb
# (per constellation + per satellite).
sat_signals = "notebooks/00_uirapuru_first_run/signals.json"

[solar_flux]
# Solar flux provider: "srd" (SRD/SRS, WDC/BOM) or "nro" (NoRP/NRO, XDR).
source = "nro"
```

```
── estimate (phase 1) ──
  instants        : 721
  benchmark (s)   : [39.81, 31.78, 33.5] (n=[6, 12, 24])
  per sample      : 6.6348 s
  estimated time  : 79.73 min
  projected output: 28.2 GB

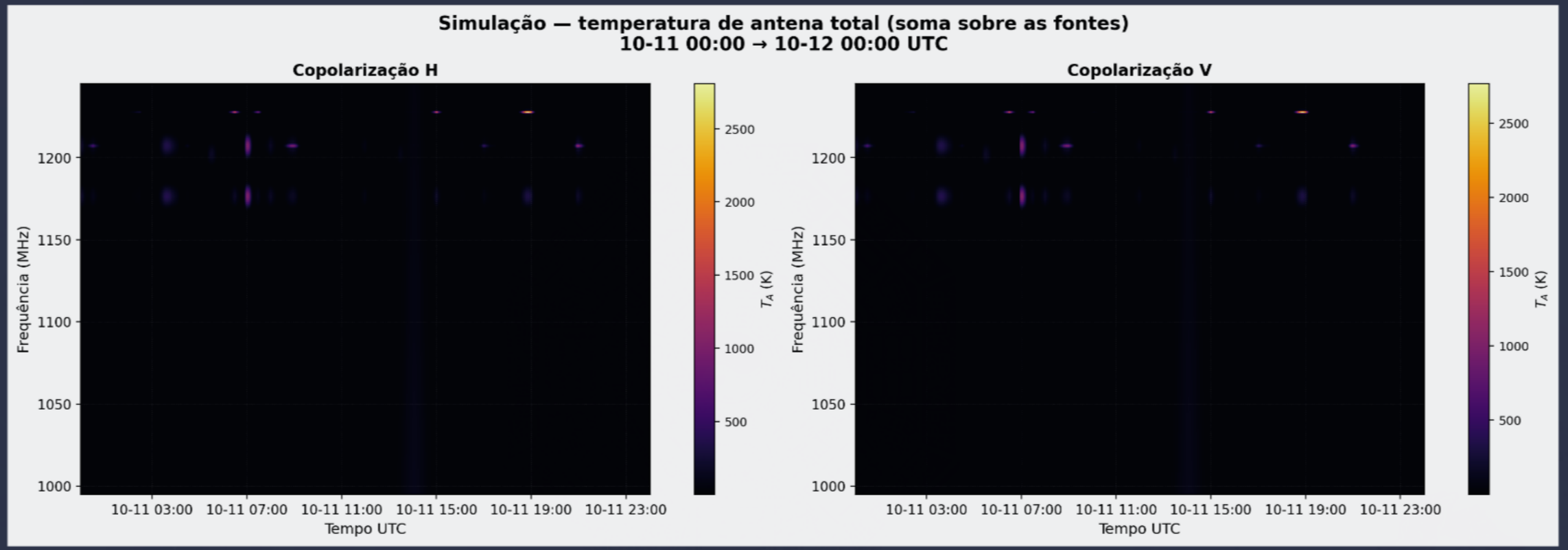
── run (phase 2) ──
  dimensions      : {'time': 721, 'frequency': 4096, 'polarization': 4, 'source': 149, 'pointing': 1}
  computation time: 1251 s (20.9 min)
  memory          : {'loaded_mb': 0.0, 'unloaded_mb': 13428.625}
  file            : 26.2 GB

── provenance (global NetCDF attrs) ──
  description = "Time-Ordered Data"
  contributions = "satellite"
```

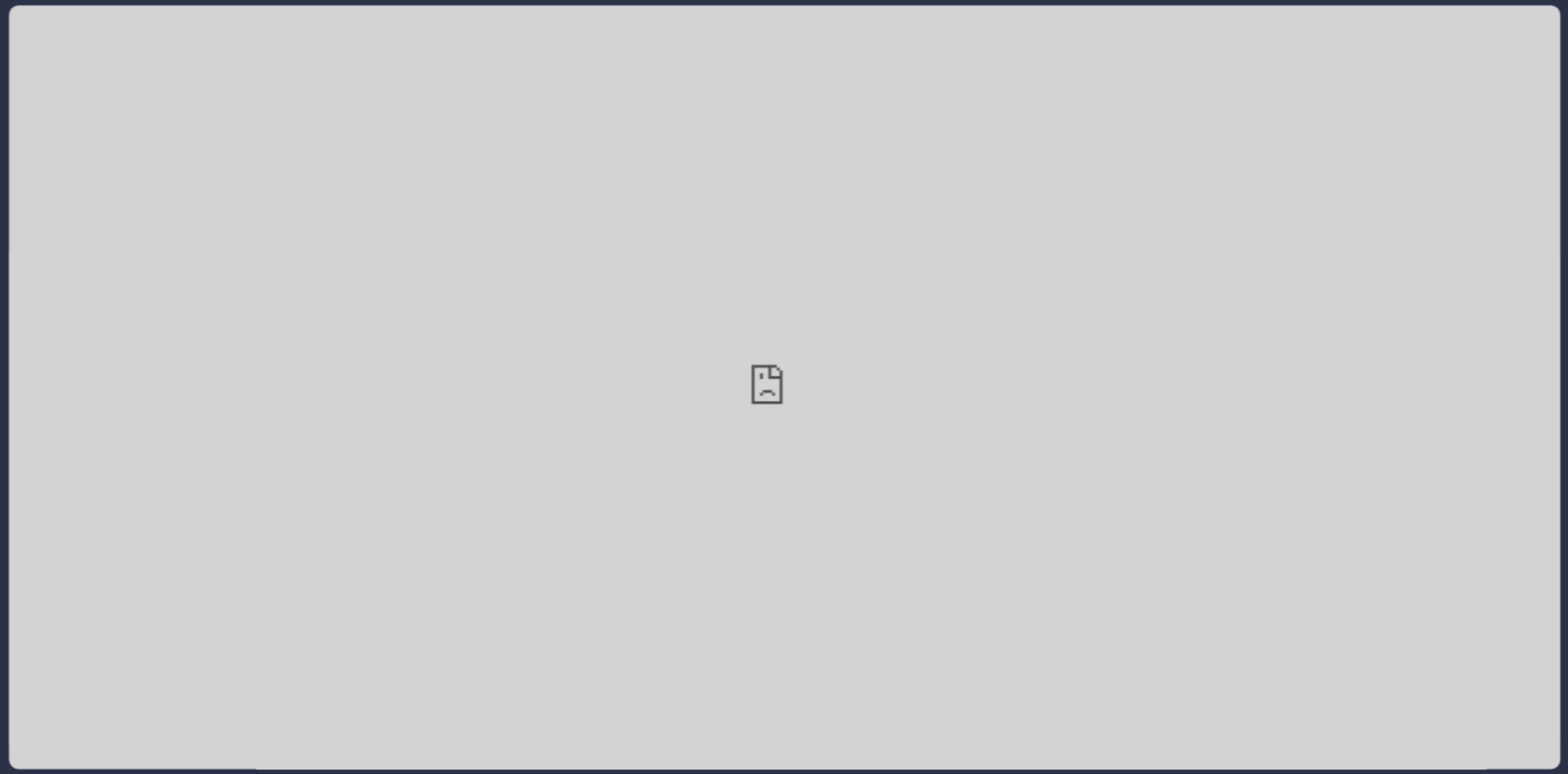
Some Results






Simulação — temperatura de antena total (soma sobre as fontes)
10-11 00:00 → 10-12 00:00 UTC
Copolarização H
Copolarização V
Frequência (MHz)
1200
1150
1100
1050
1000
10-11 03:00
10-11 07:00
10-11 11:00
10-11 15:00
10-11 19:00
10-11 23:00
Tempo UTC
2500
2000
1500
1000
500
T_A (K)

Waterfall Grid (interactive)



Future Work

