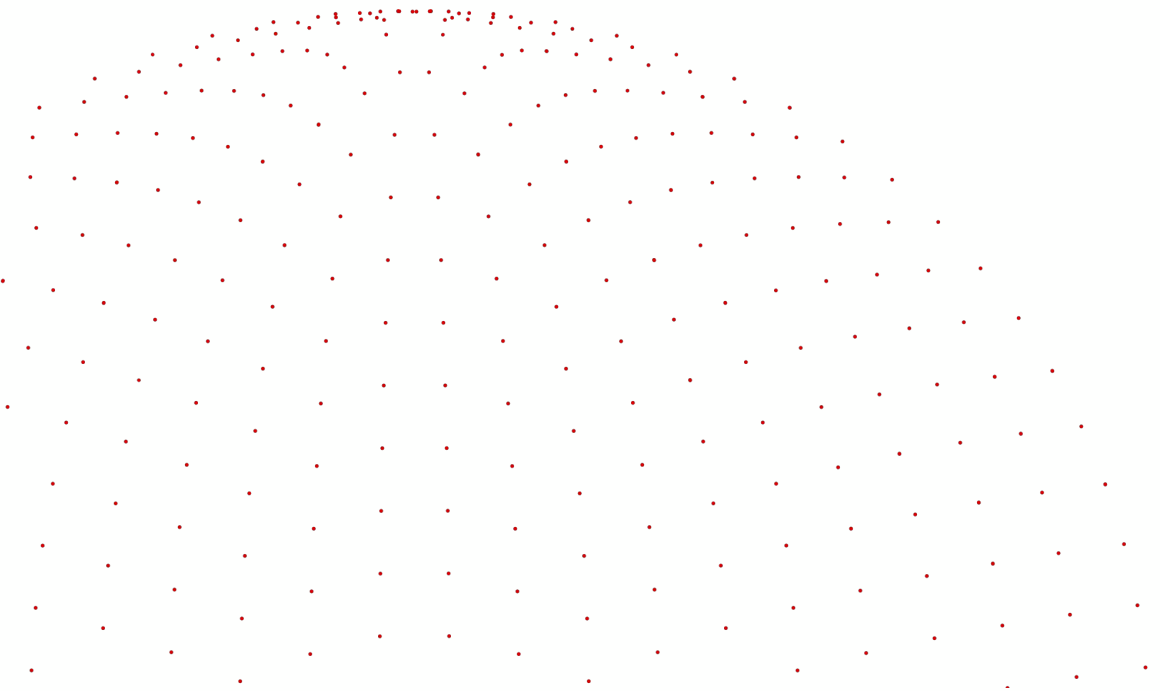
spire

Constellation Modeling, Performance Prediction, and Operations Management for the Spire Constellation

SSC21-I-13

Jeroen Cappaert
Chief Technology Officer / Co-Founder

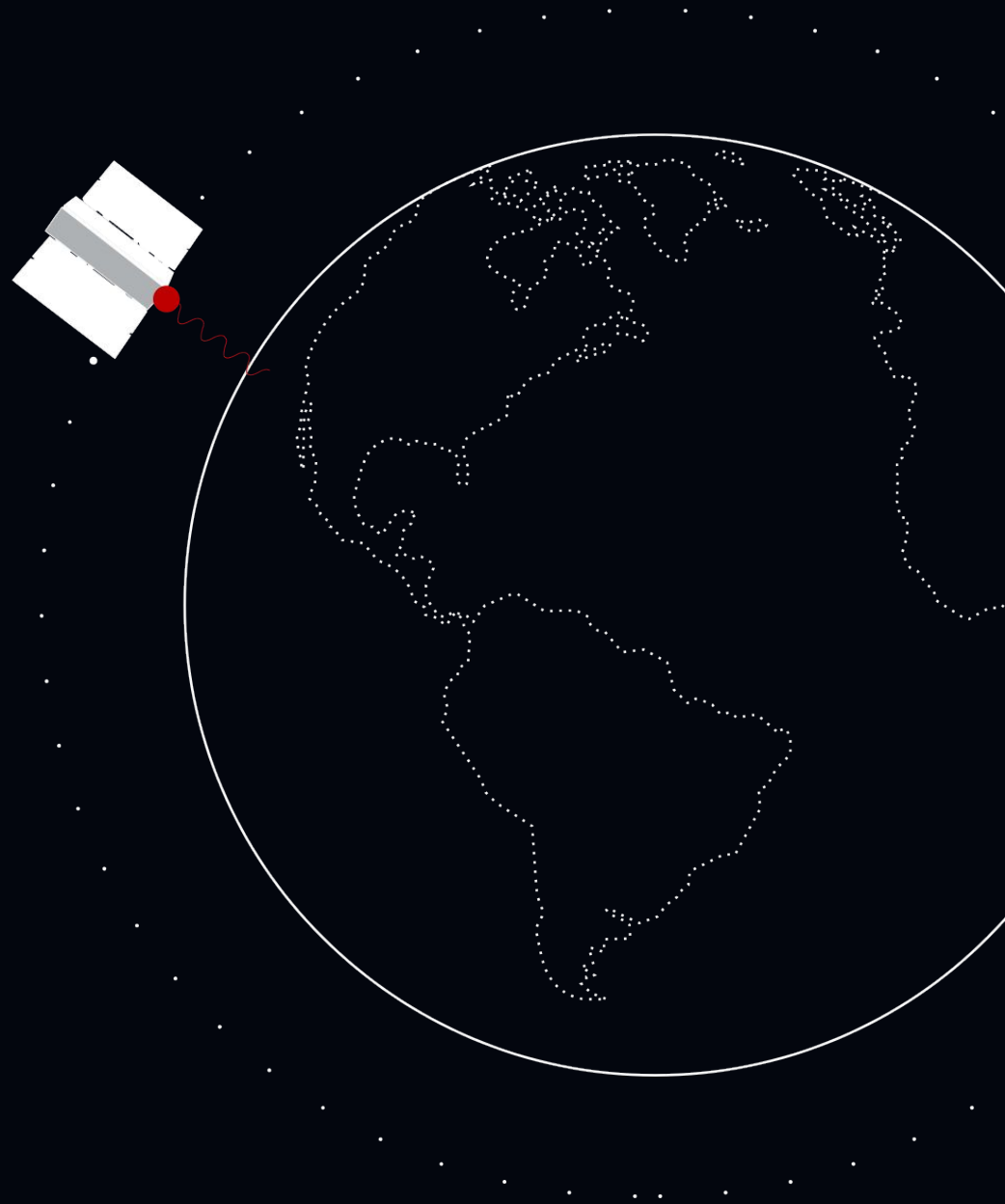
Agenda



1. Introduction
2. Constellation Modeling and Performance Prediction
3. Constellation Operational Planning (The Optimizer)
4. Performance Monitoring
5. Enabling Customers with Standardized User Services
6. Conclusions

spire

Introduction

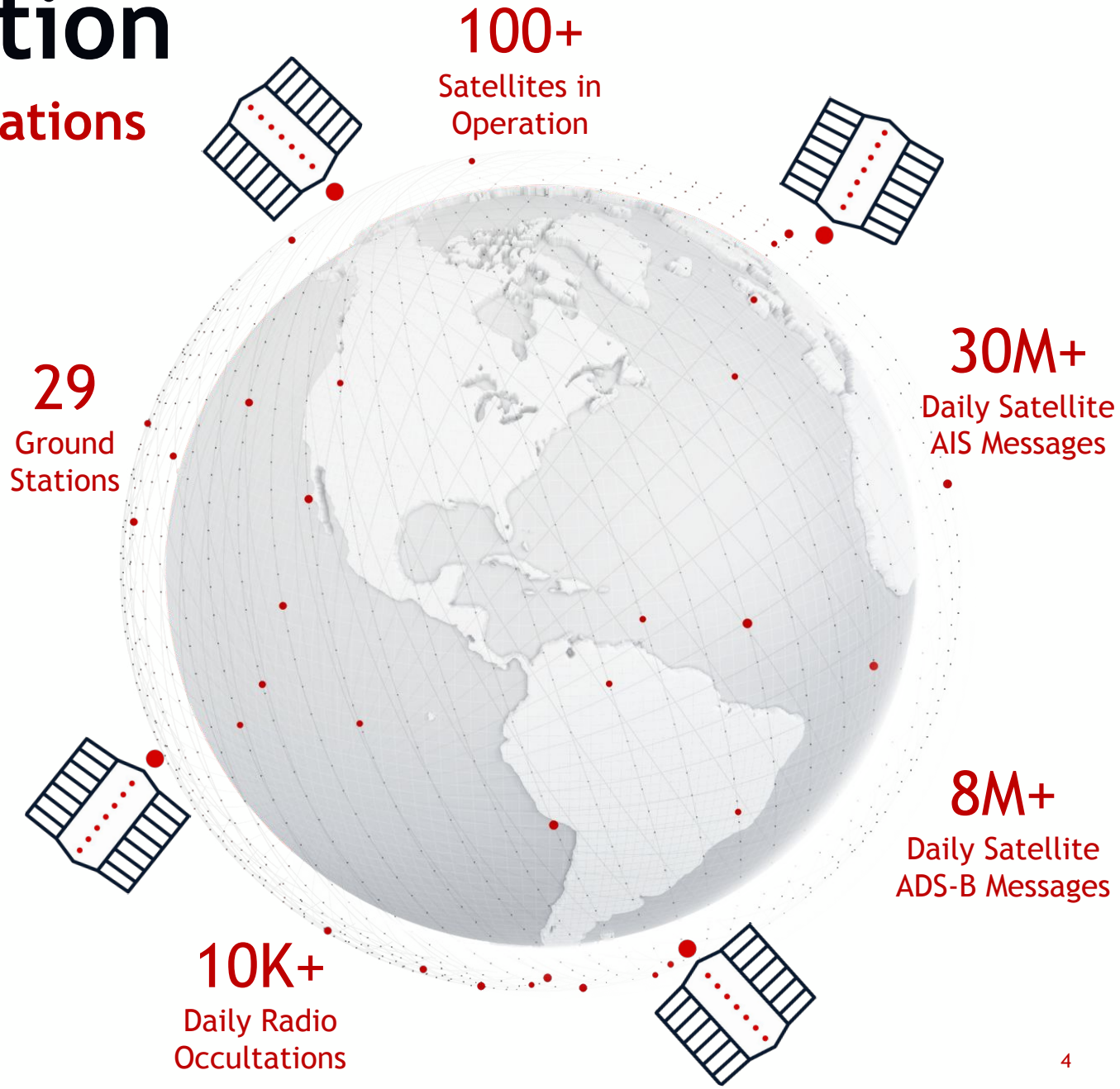


spire

The Spire Constellation

One of the largest private constellations in the world

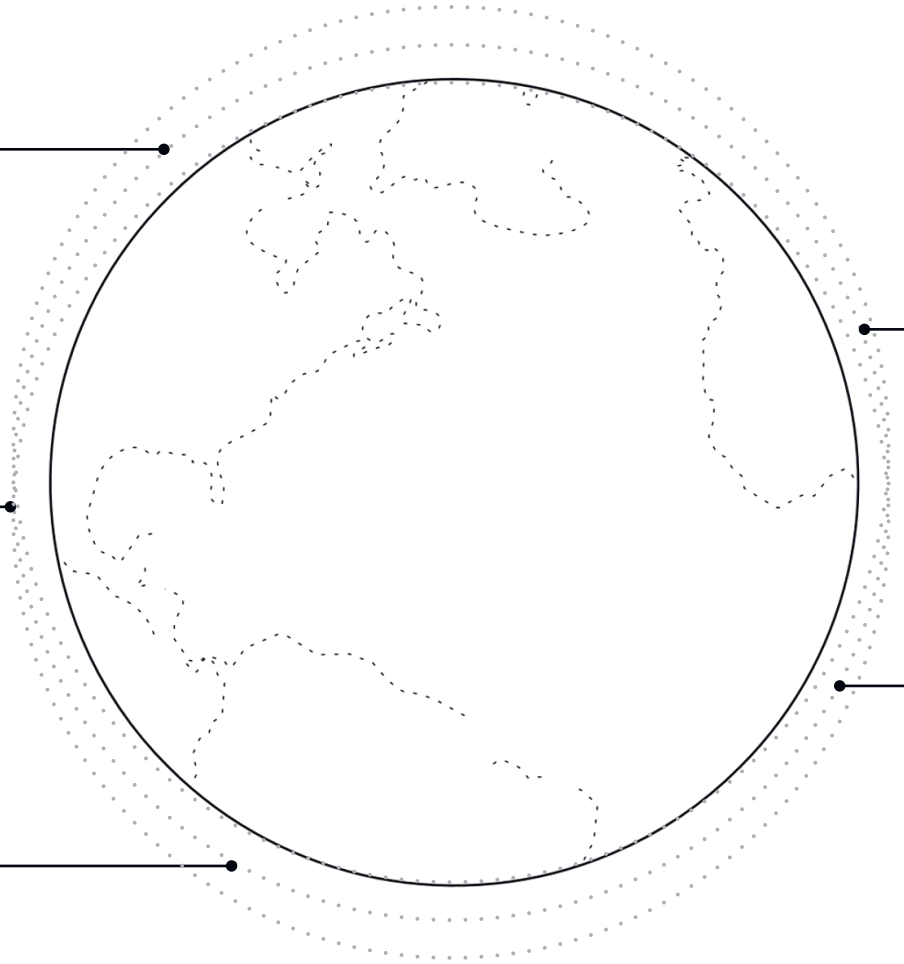
- The LEMUR is Spire's 3U CubeSat platform used to track maritime, aviation, weather and other activity from space
- We operate one of the largest RF sensing fleets and are one the largest producers of radio occultation and space weather data
- Our data provides a global view with coverage in remote regions like oceans and poles
- We are continuously launching improved sensors and upgrading them in-orbit
- We turn ideas into live feed from space in as little as 6-12 months



spire

The Spire Constellation

114 LEO-Based Cubesats in a diverse set of orbits enabling global coverage high daily revisit



Polar - Sun Synchronous Orbit
76 Satellites

Equatorial Orbit
4 Satellites

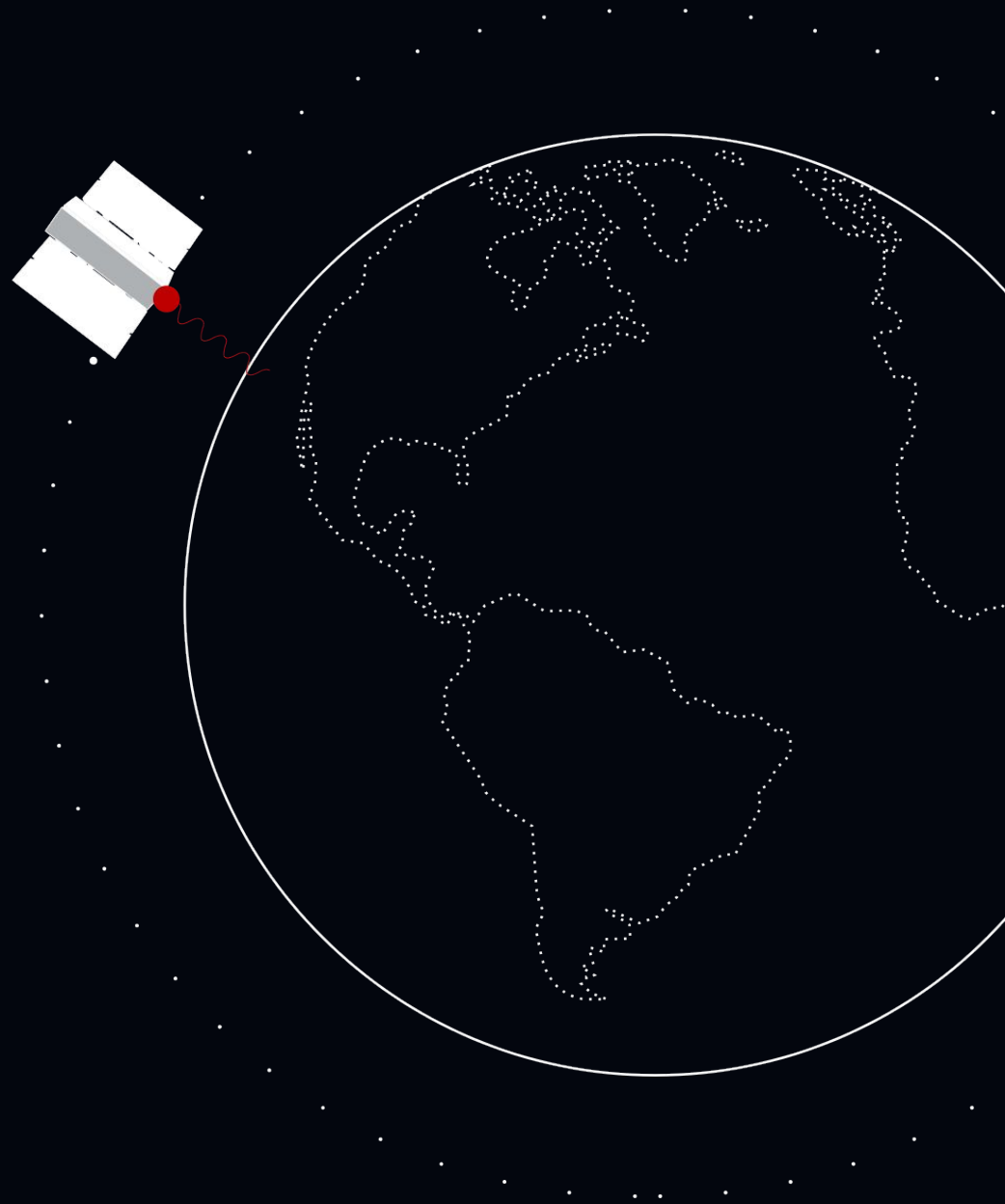
37 Degrees Orbit
8 Satellites

83-85 Degree Orbit
4 Satellites

ISS 51.6 Degree Orbit
22 Satellites

spire

Constellation Modeling and Performance Prediction

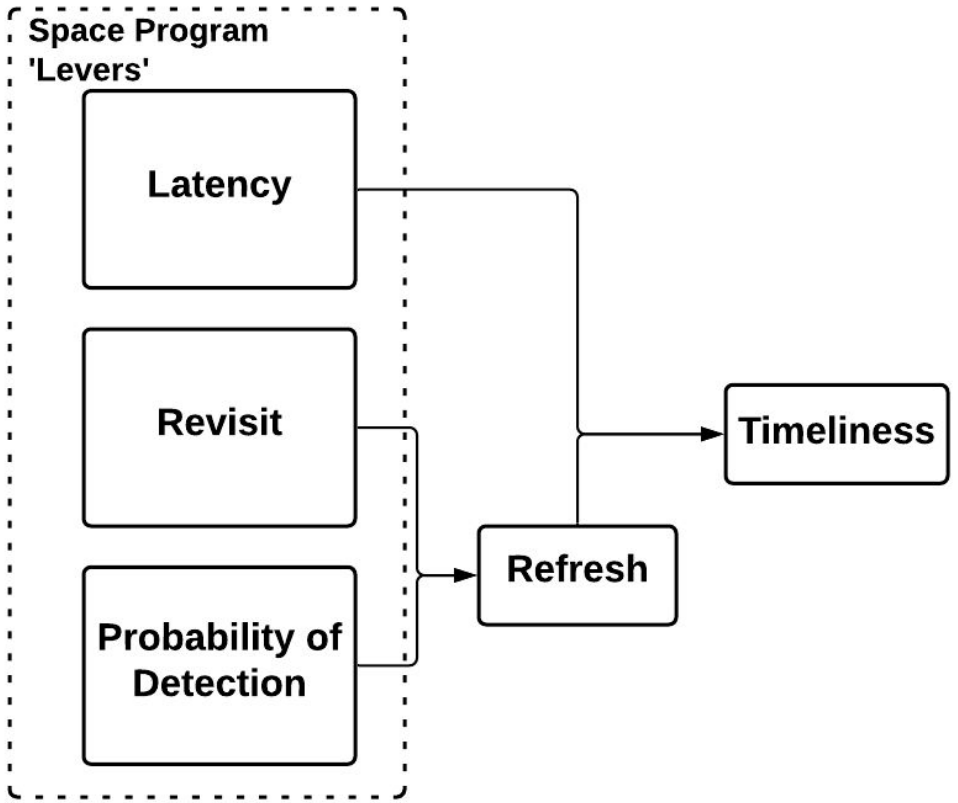


Constellation Modeling

Modeling translates business requirements into engineering requirements

Customers impose unique sets of requirements based on characteristics of the data that must be produced:

- **Latency**: the time delay from data collection to customer availability through an API
- **Revisit**: the time between two consecutive observations of a specific point on Earth
- **Probability of Detection**: how likely a satellite is to successfully collect the desired data for a specific point
- **Refresh**: revisit + probability
- **Timeliness**: refresh + latency



Measuring On-Orbit Data Collection

spire

Constellation Modeling

Details about existing and future satellites and ground stations are required in order to define a candidate constellation:

- TLEs
- Hardware version
- Payloads (footprint shape, a fixed or variable data collection rate, the priority of the data, and the probability of detection)
- Communications configurations (type of contact, geometric requirements, regulatory constraints, and data rates)
- Scheduling constraints (e.g. blackout windows)

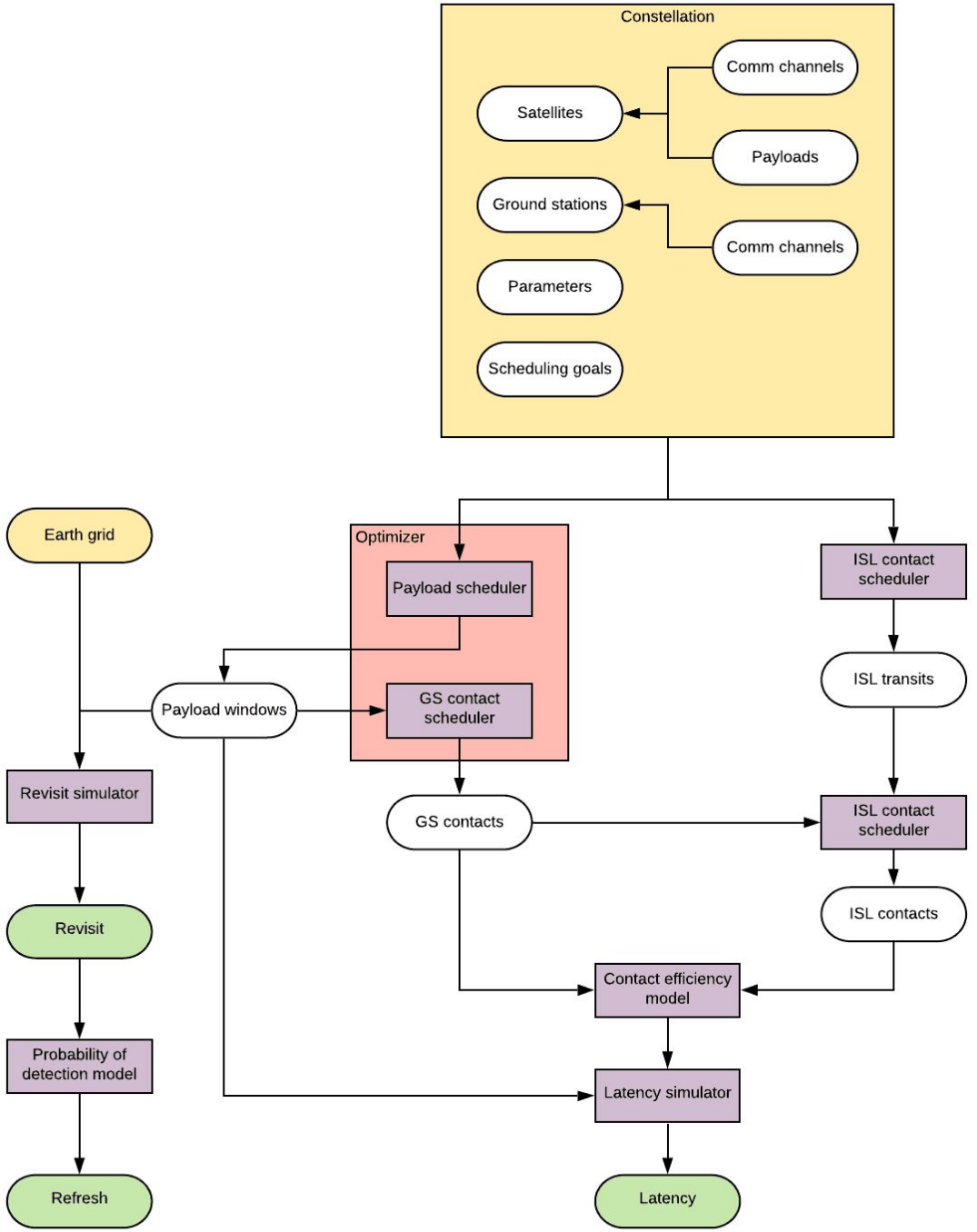
The constellation model simulator is provided the following inputs:

- Customer-provided objectives (e.g. x MB or ADS-B data < 15min latency over open ocean)
- A set of parameters for the models (e.g. 30s of handshaking time per contact etc.)
- Start and end times

Model Architecture

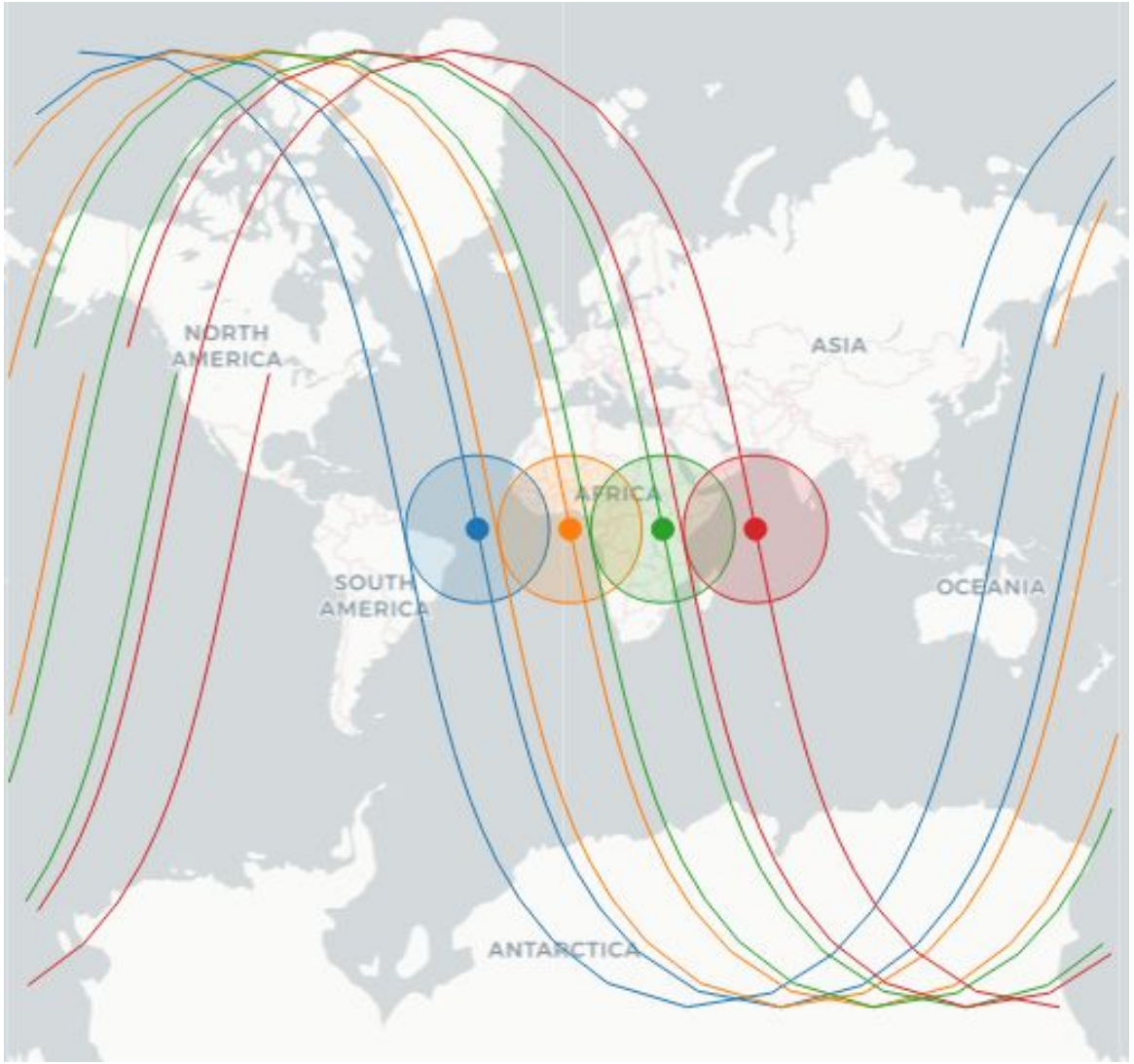
The constellation model architecture can be divided into 2 parts

- **Revisit and Refresh** inputs include:
  - User-defined Earth grid with points at which the revisit rate will be calculated
  - Optimizer payload schedule
  - Satellite details
- **Latency** inputs include:
  - Optimizer payload schedule
  - Optimizer ground station contact schedule
  - ISL contact schedule



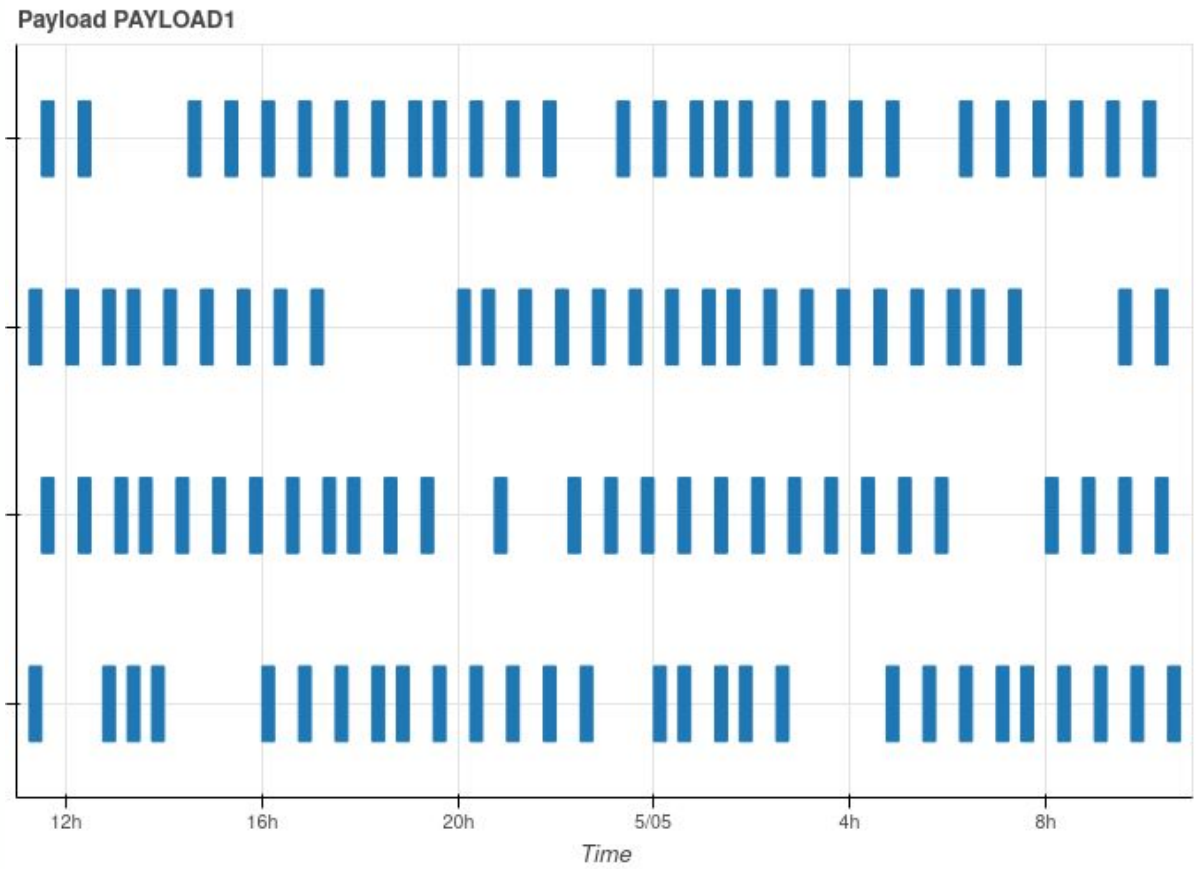
Use Case Example

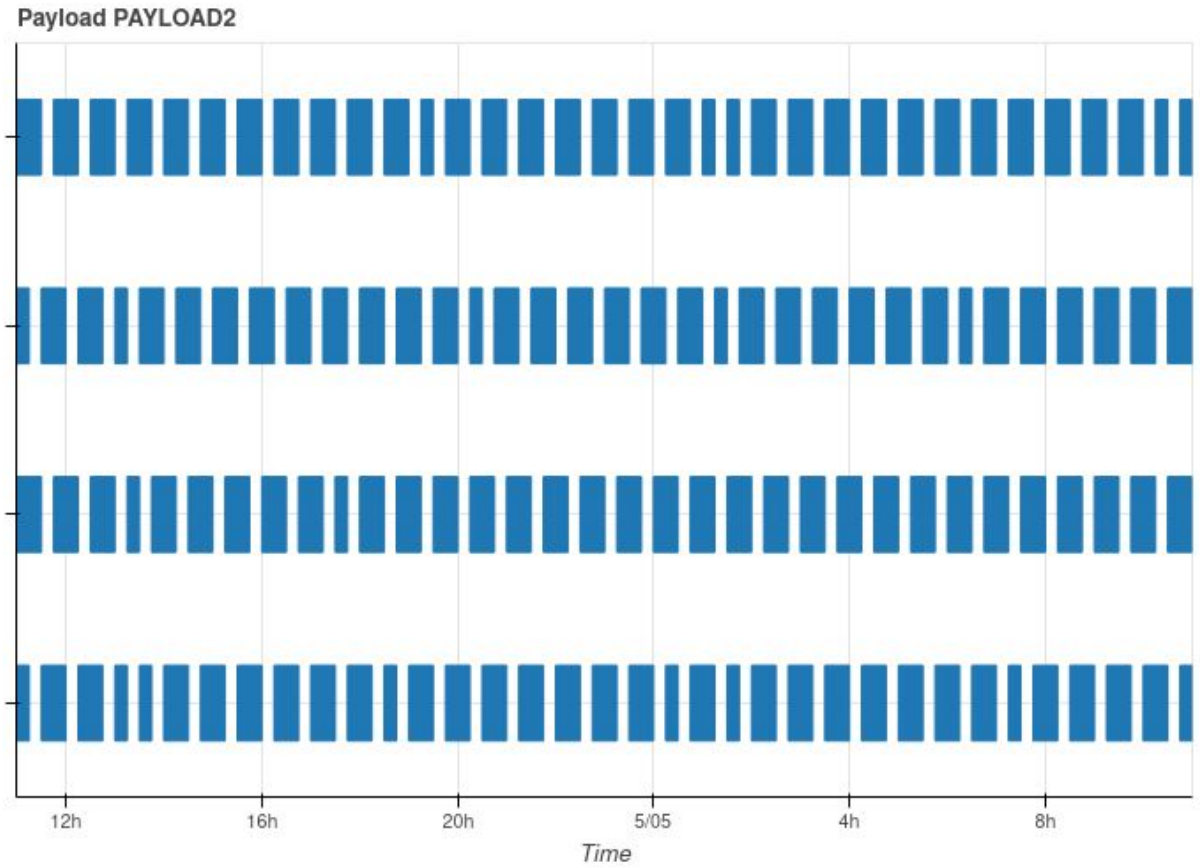
- 4 ISL satellites
- Sun-Synchronous Orbit planes with a right ascension of the ascending node increment of 30 degrees and with the same true anomaly
- 9 ground stations
- 2 payloads per satellite: PAYLOAD-1 and PAYLOAD-2 which cannot be scheduled at the same time



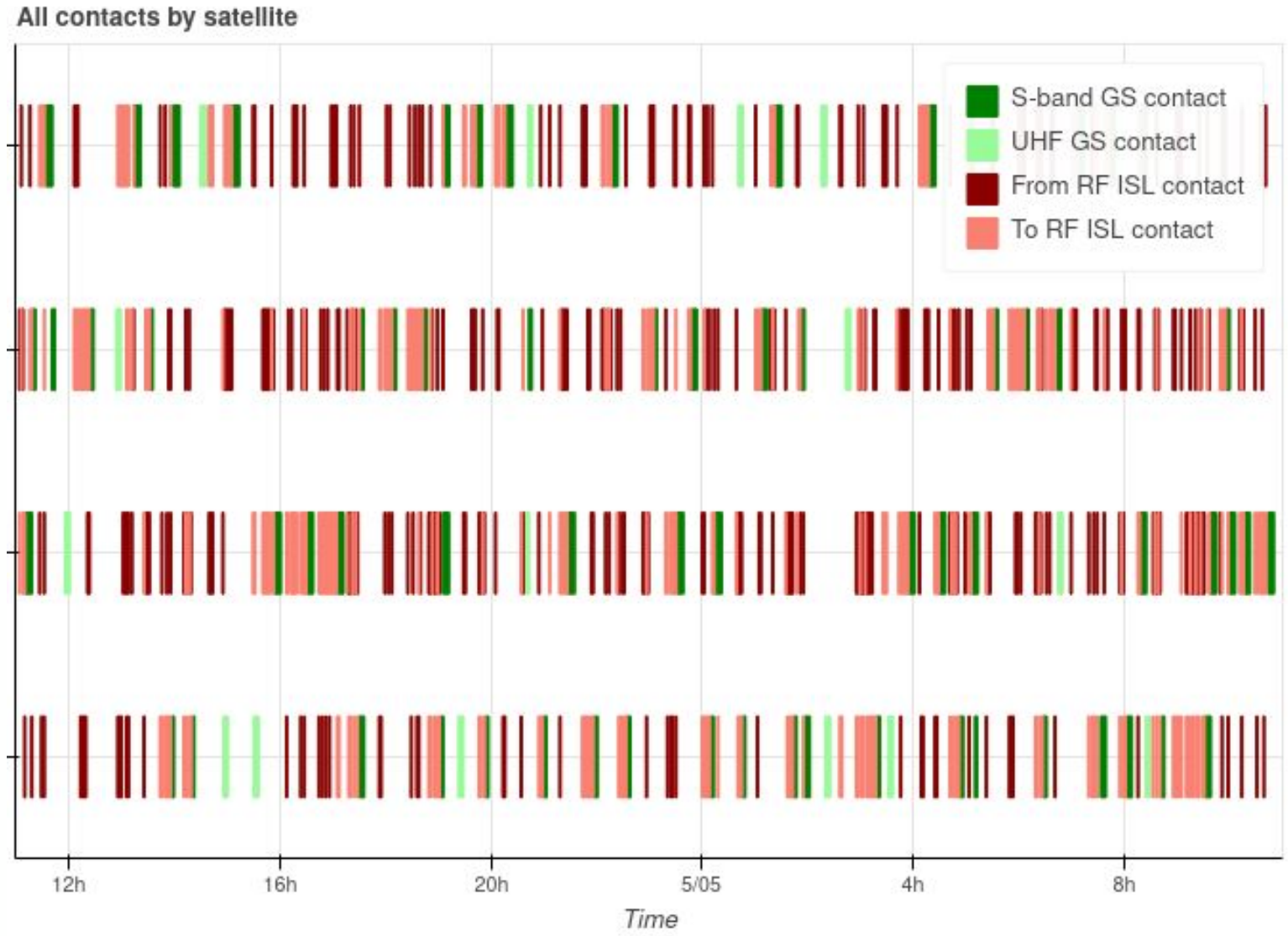
Orbits of the 4 satellites

Payload Schedules

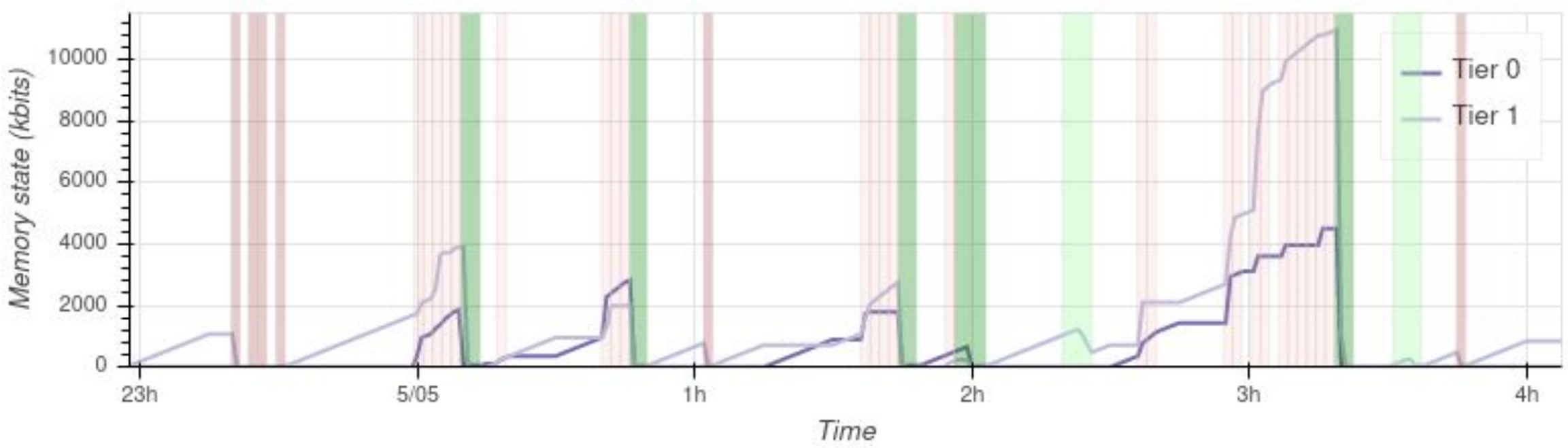




GS/ISL Contact Schedule

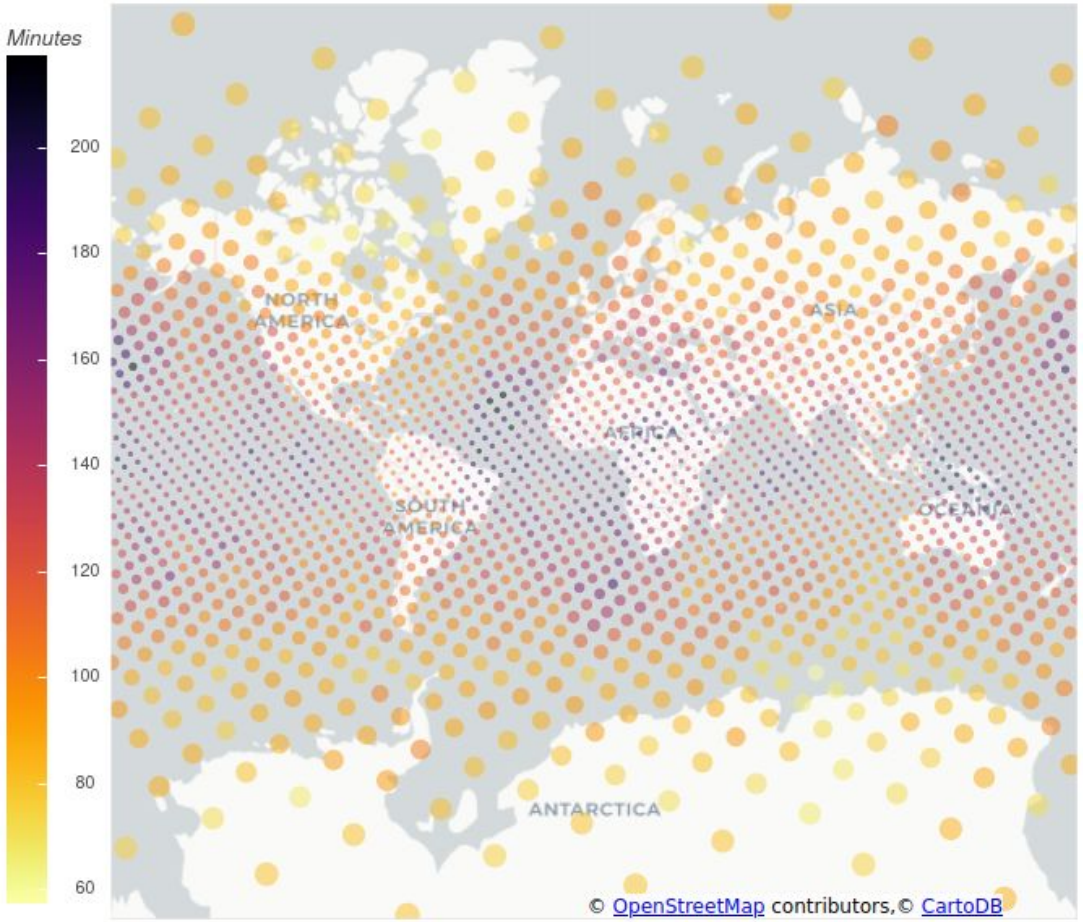


Satellite Memory State

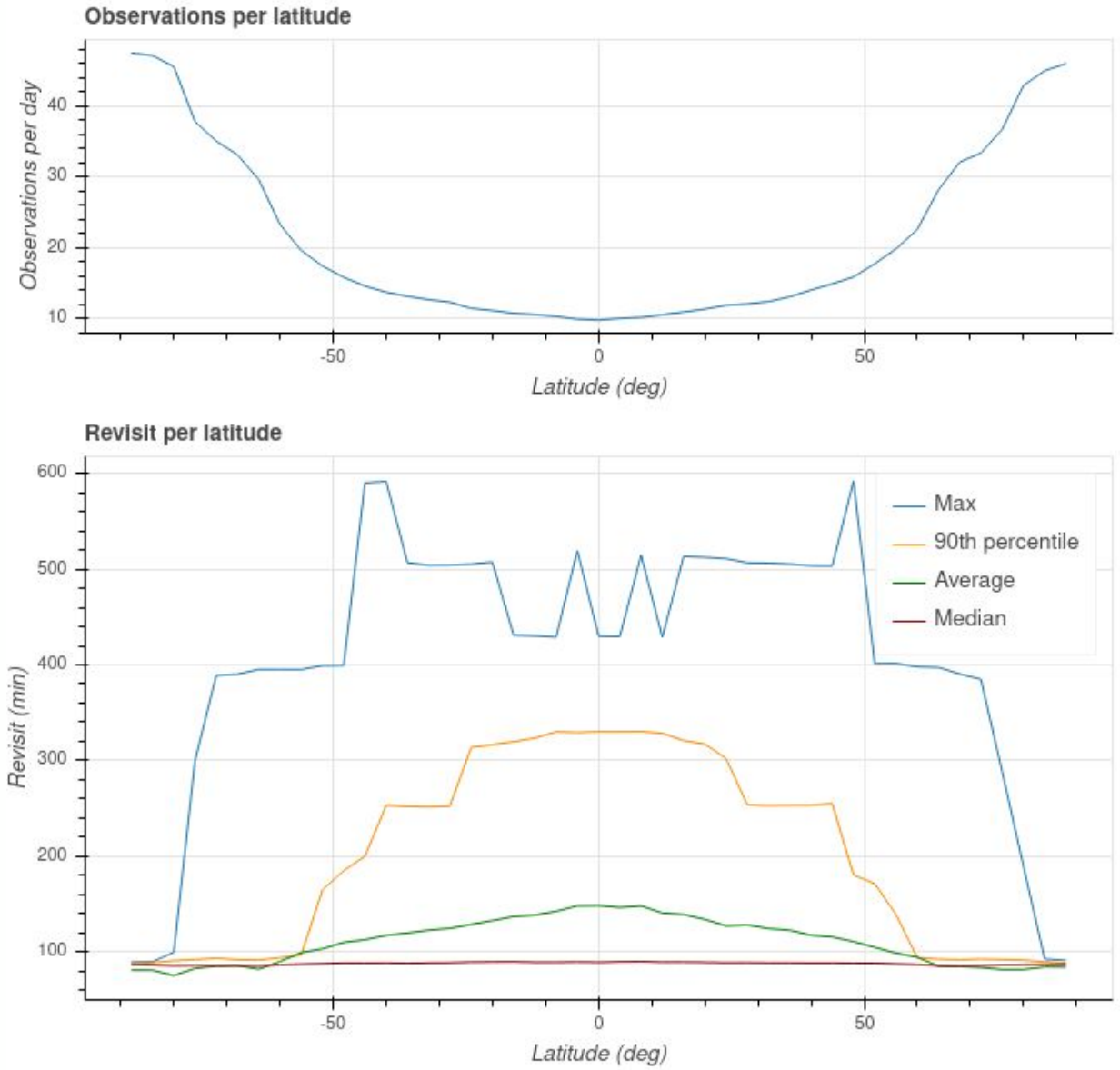


The amount of data in each data tier that the satellite stores throughout the duration of the simulation (Tier 0 is higher priority than Tier 1)

Revisit Rates



Average Revisit Heatmap



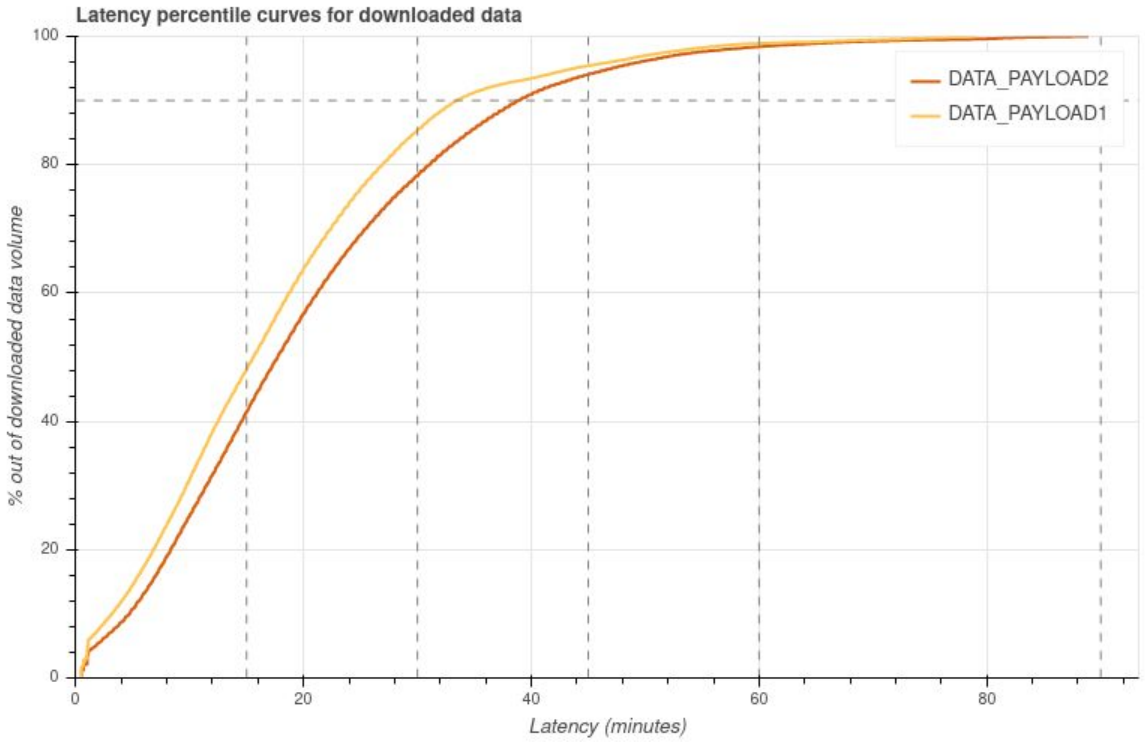
Revisit Statistics by Latitude

Latency Curves

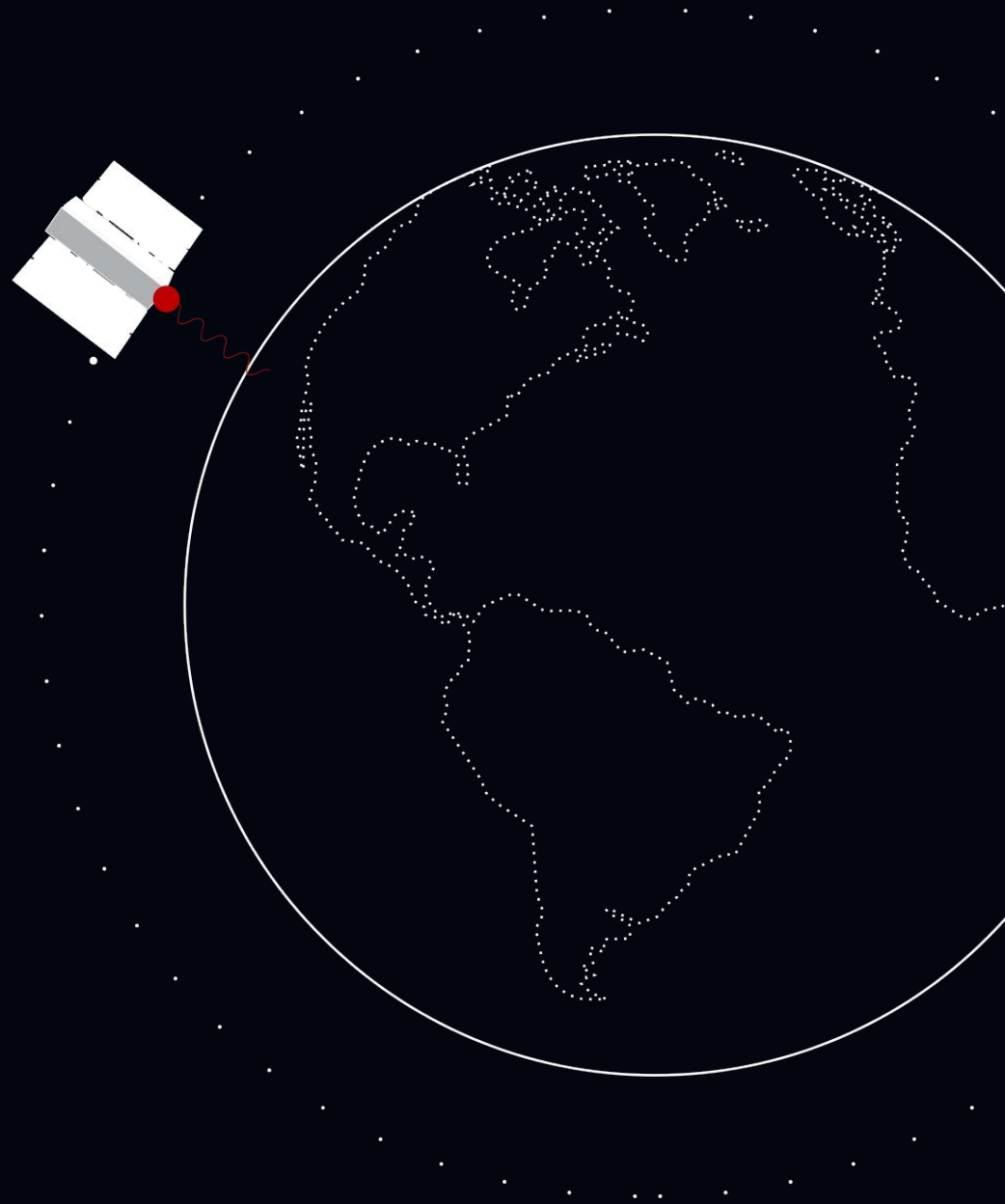
Latency percentile curves allow analysis of the downloaded data

The y-axis shows the percentage of data that has a latency below the x-axis value

Each payload in the simulation has an associated curve: here PAYLOAD-1 has a better latency because its data priority level is higher than PAYLOAD-2



Constellation Operational Planning (The Optimizer)



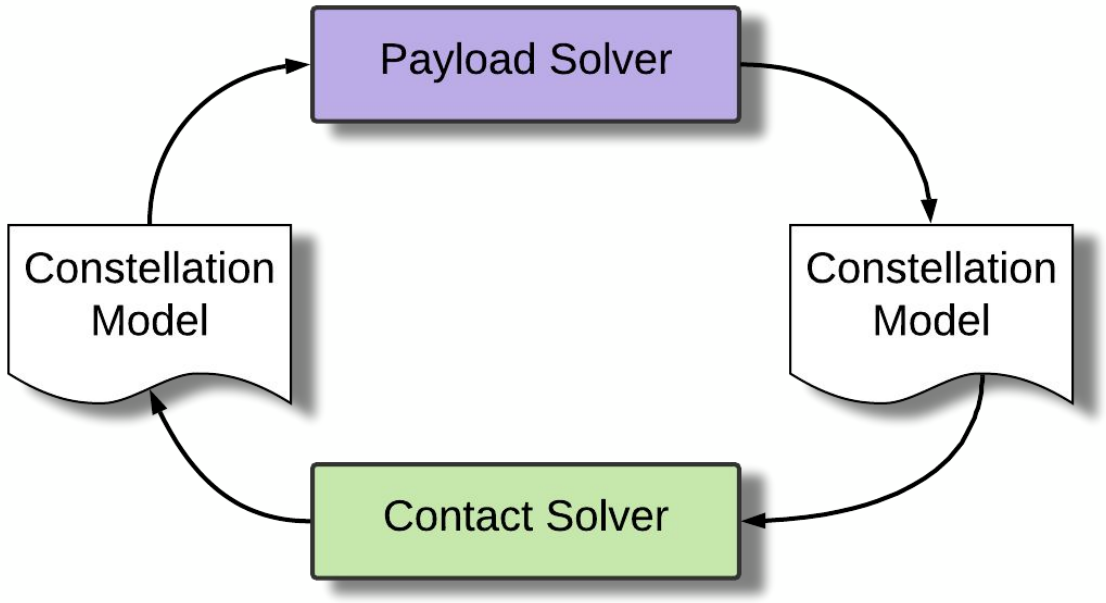
spire

Model Design

All schedulers work with a single model format, known as the Unified Model, which encapsulates all information required to represent the constellation at a set time

The Unified Model is the standard format for both inputs and outputs across Spire's solvers, empowering uses and enabling long-duration simulations

Using the communications configuration model, the Unified Model is able to represent constraints on what, when, and how assets can communicate



The Unified Model

Payload Scheduling

SPORE, the Spire Payload Scheduler is an advanced, fully-automated, satellite constellation scheduling system in active production use

It schedules optimized data collections for the entire constellation by performing payload deconfliction, area of interest targeting, and general power management (through flexible duty ratio constraints)

SPORE incorporates information on each individual satellite's power, coverage, and conflicting hardware subsystems to balance collections across time

Example: Payload schedules de-prioritizing collection over the Atlantic

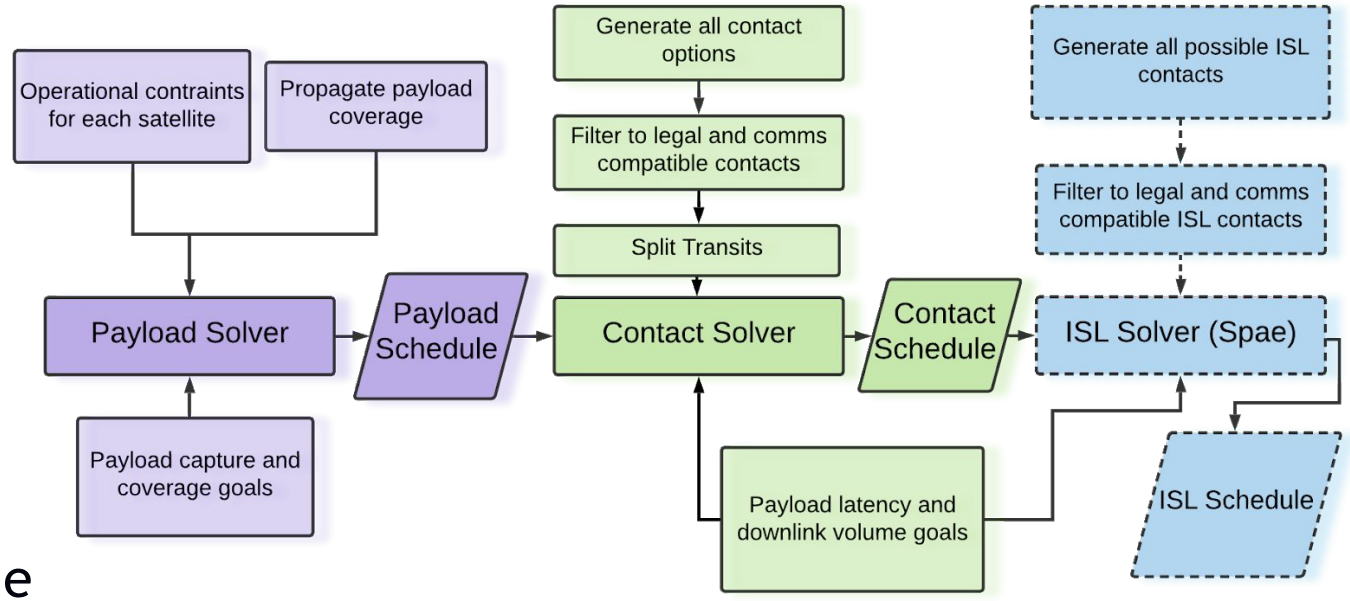
Contact Scheduling

The payload schedule output from SPORE is fed into the Spire Contact Solver, which selects an optimal set of accesses to ensure payload data is downlinked efficiently

The Contact Solver is self-aware, accounts for constraints, and can schedule the entire constellation in less than 5 minutes

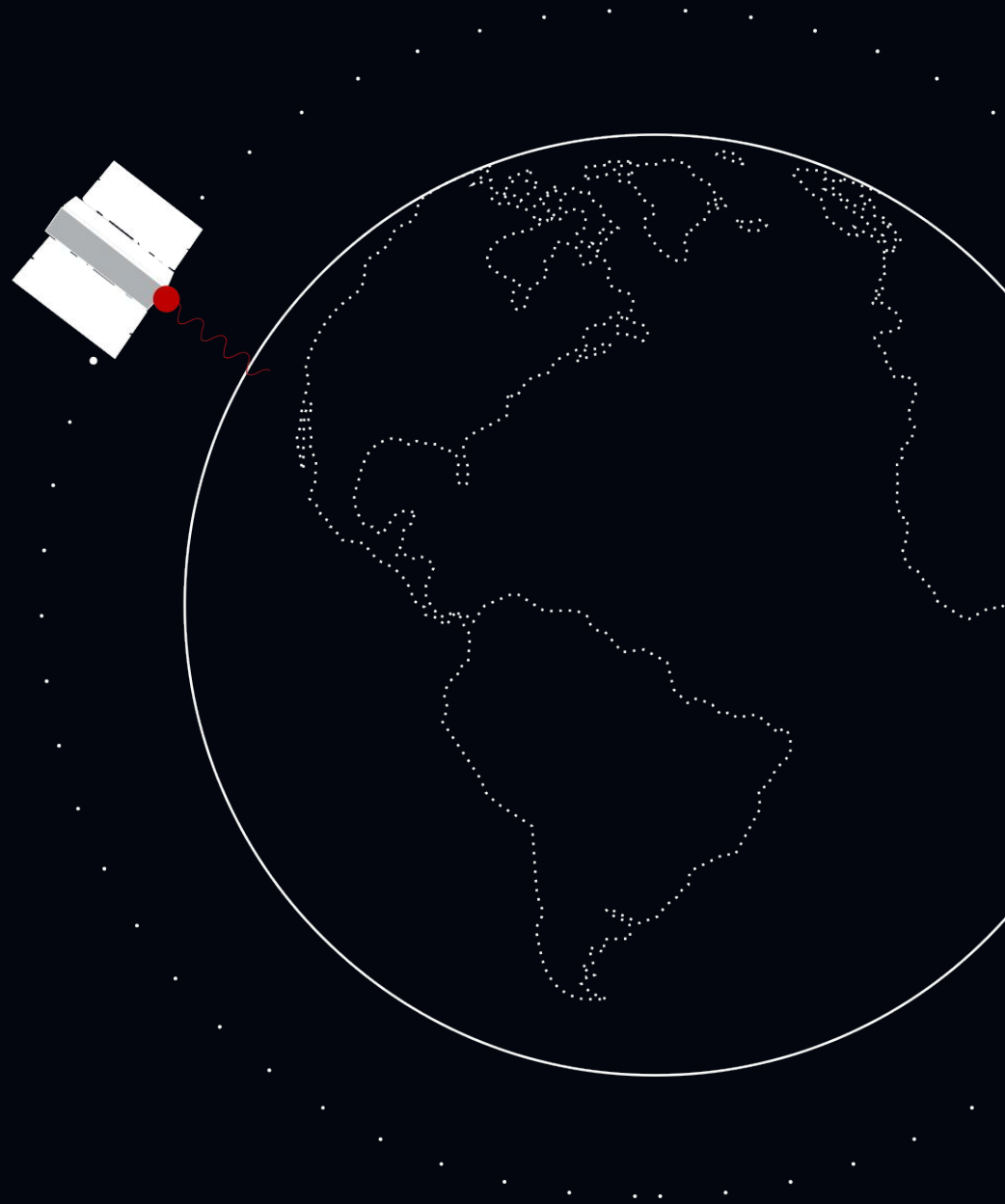
The contact schedule is optimized for business objectives and contact time is allocated to fleet assets based on the volume and value of data collected

Spire schedulers are intent-driven and operate on declaratively-defined models



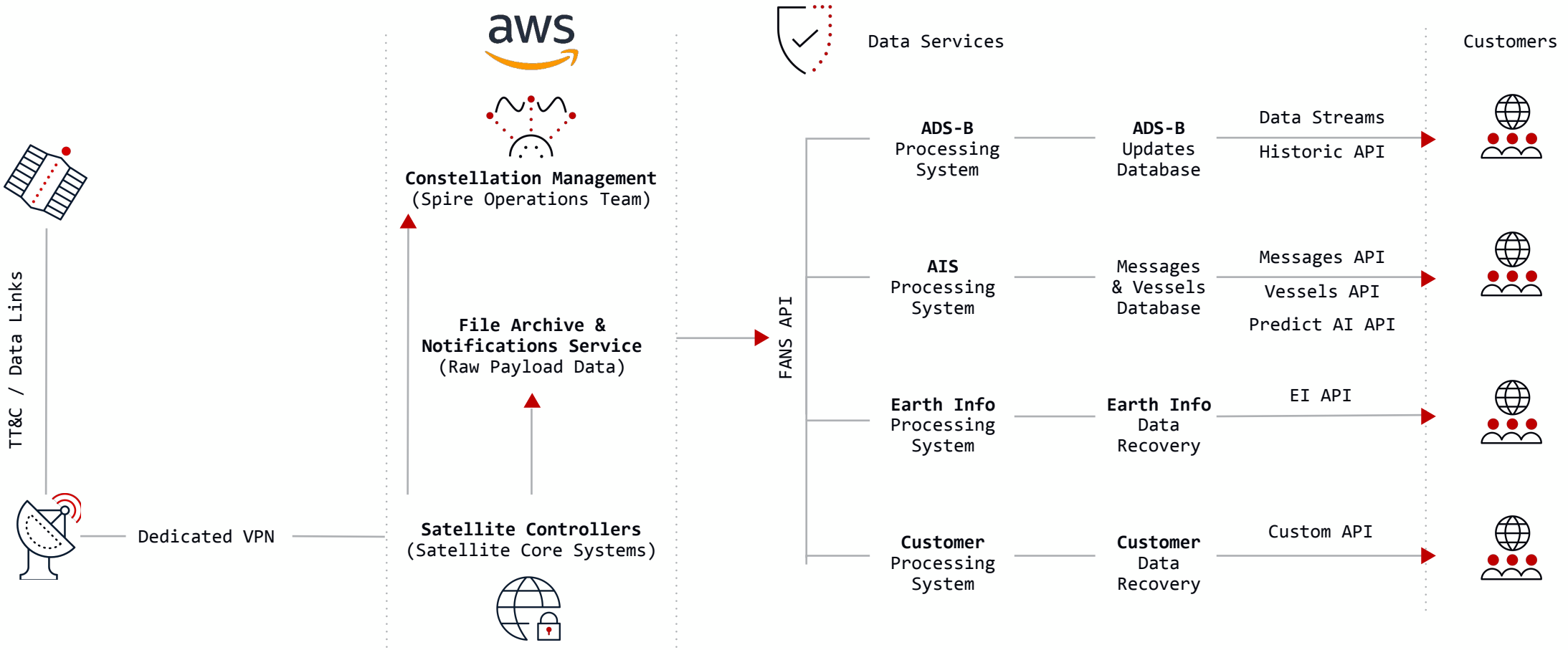
The Constellation Model

Performance Monitoring



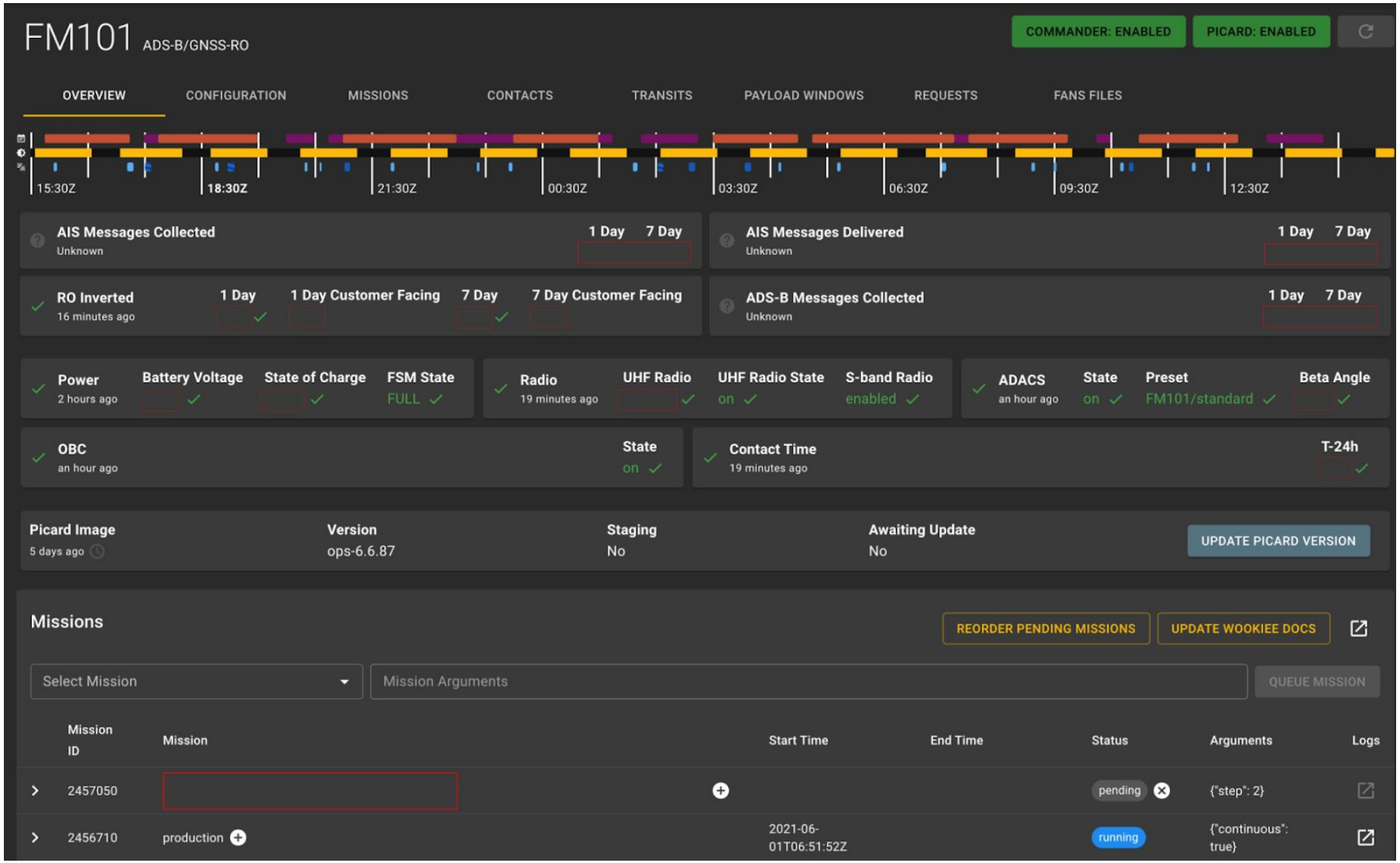
spire

Scalable and Secure Data Services Pipeline

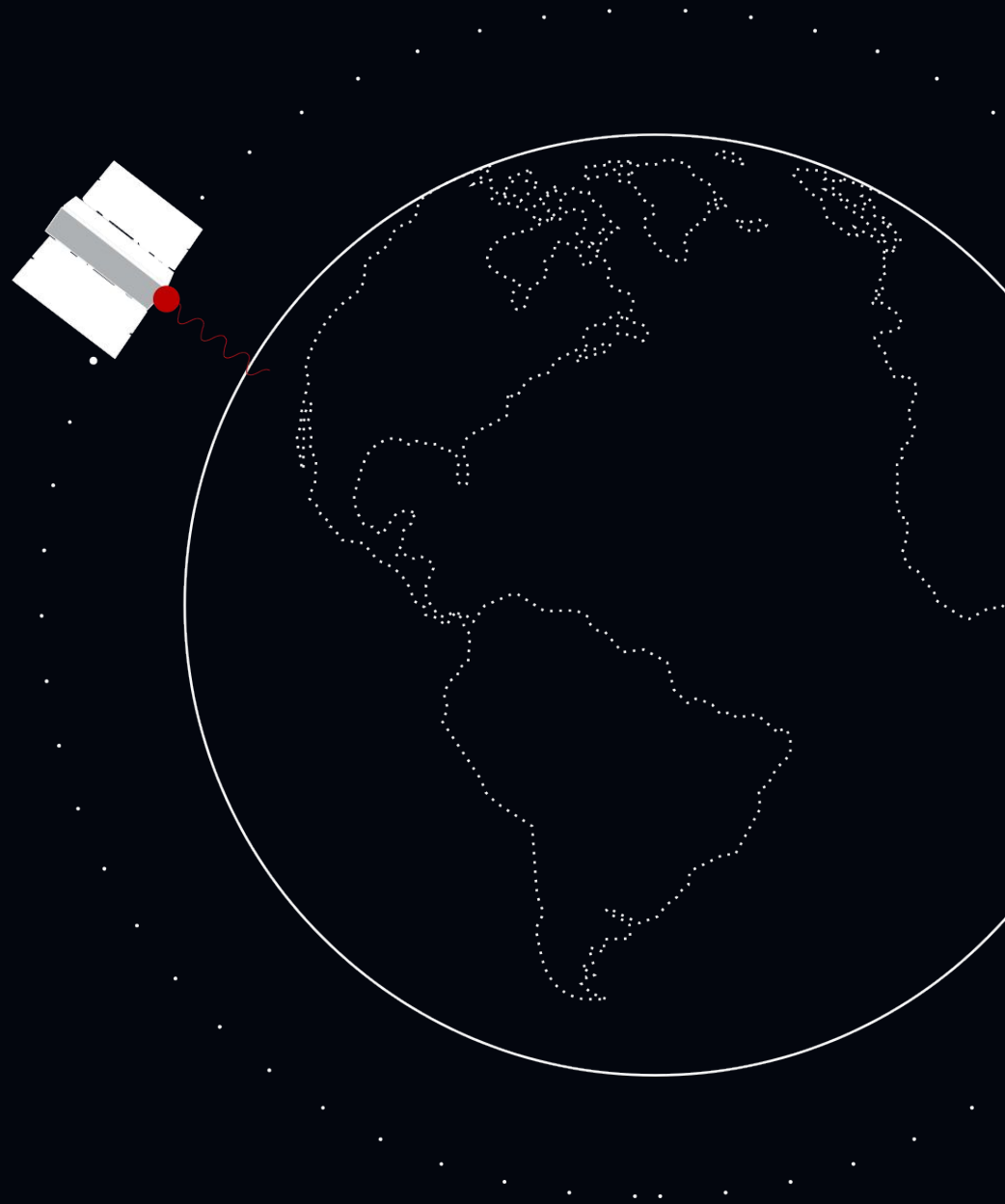


Fully secure and encrypted TT&C link, data links, U.S.-based cloud processing, and data distribution service; data remains encrypted from collection onboard the satellite to desktop user token

Spire Operations Center



Enabling Customers with Standardized User Services



spire

Spire User Services

Scheduling API

Provides access to the payload scheduler (SPORE) within the Optimizer system

Allows customers to generate candidate payload windows automatically, mirroring how Spire schedules for the full constellation

The generated payload windows can be submitted to the Tasking API for execution on orbit

Tasking API

Provides a set of RESTful endpoints to the customer for tasking, configuring, and managing payload operations (a single payload window may consist of multiple steps)

The API can be used directly for manual scheduling of individual windows or as a service for automated system scheduling

It can also be used to upload software or other files to the customer's payload

Spire Linux Agent

Enables seamless integration with the Spire LEMUR satellite bus

Agent binaries and source code are provided to support their development and C and Python SDKs are available as interfaces

Provides access to the Data Pipeline API for customers.

Spire User Services

Data Pipeline API

Allows users to download data from their payload to their ground-based data storage in AWS S3

Designed to abstract the complications of managing a disruption-tolerant network from the end user

Provides a simple, always available to access the data pipeline

Made available by the Spire Linux Agent and associated SDKs

Signaling API

Provides payloads on the LEMUR satellite bus with the ability to receive and act on events generated by the bus

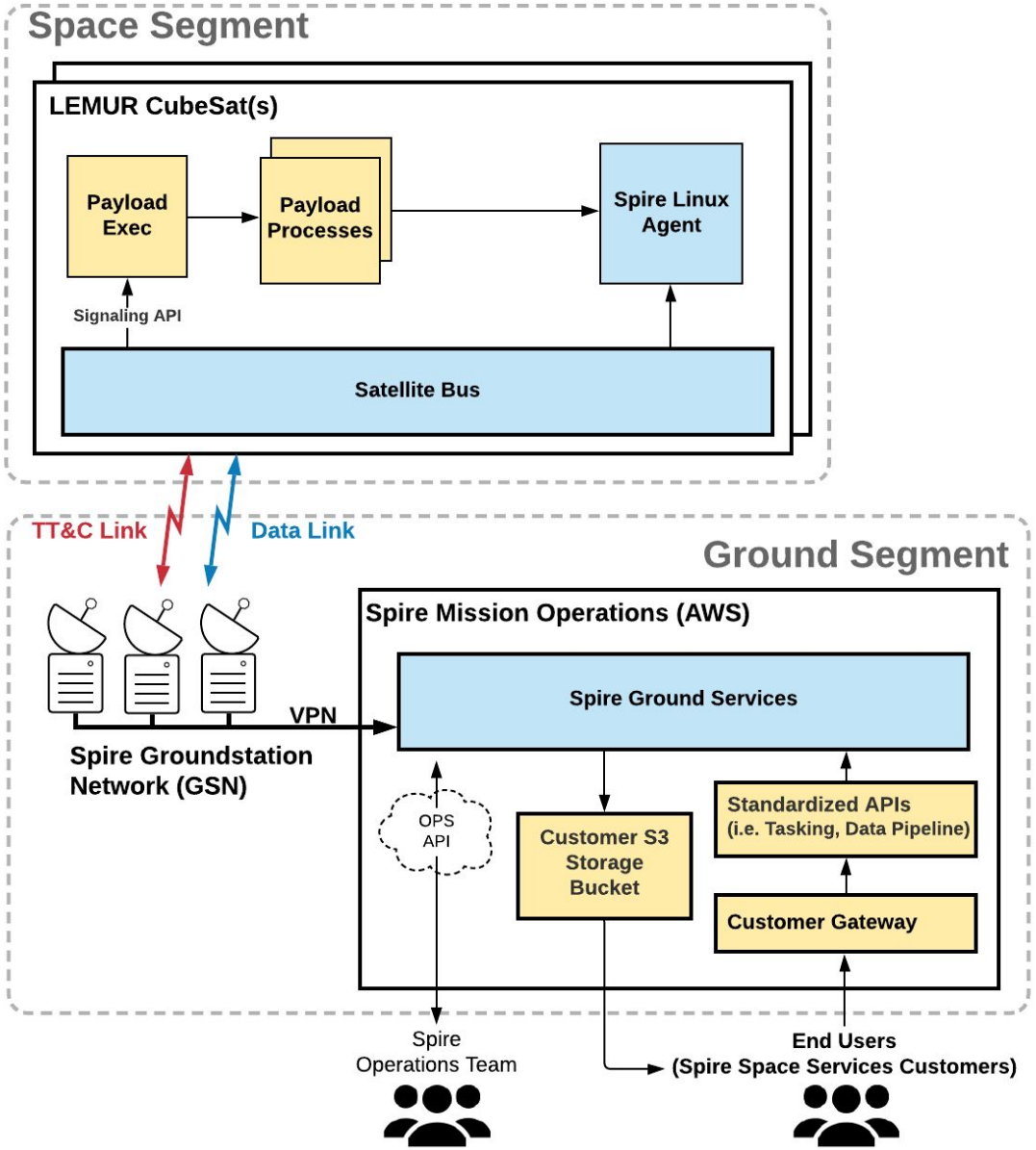
The API consists of:

- A payload executable used to respond to satellite bus events
- Conventions of where payload window configuration files and uplinked packages are placed on the payload file system by the satellite bus
- Window configuration file schemas provided by the satellite bus for payload executables to use
- Argument schemas that payload executables must accept to handle satellite bus signals

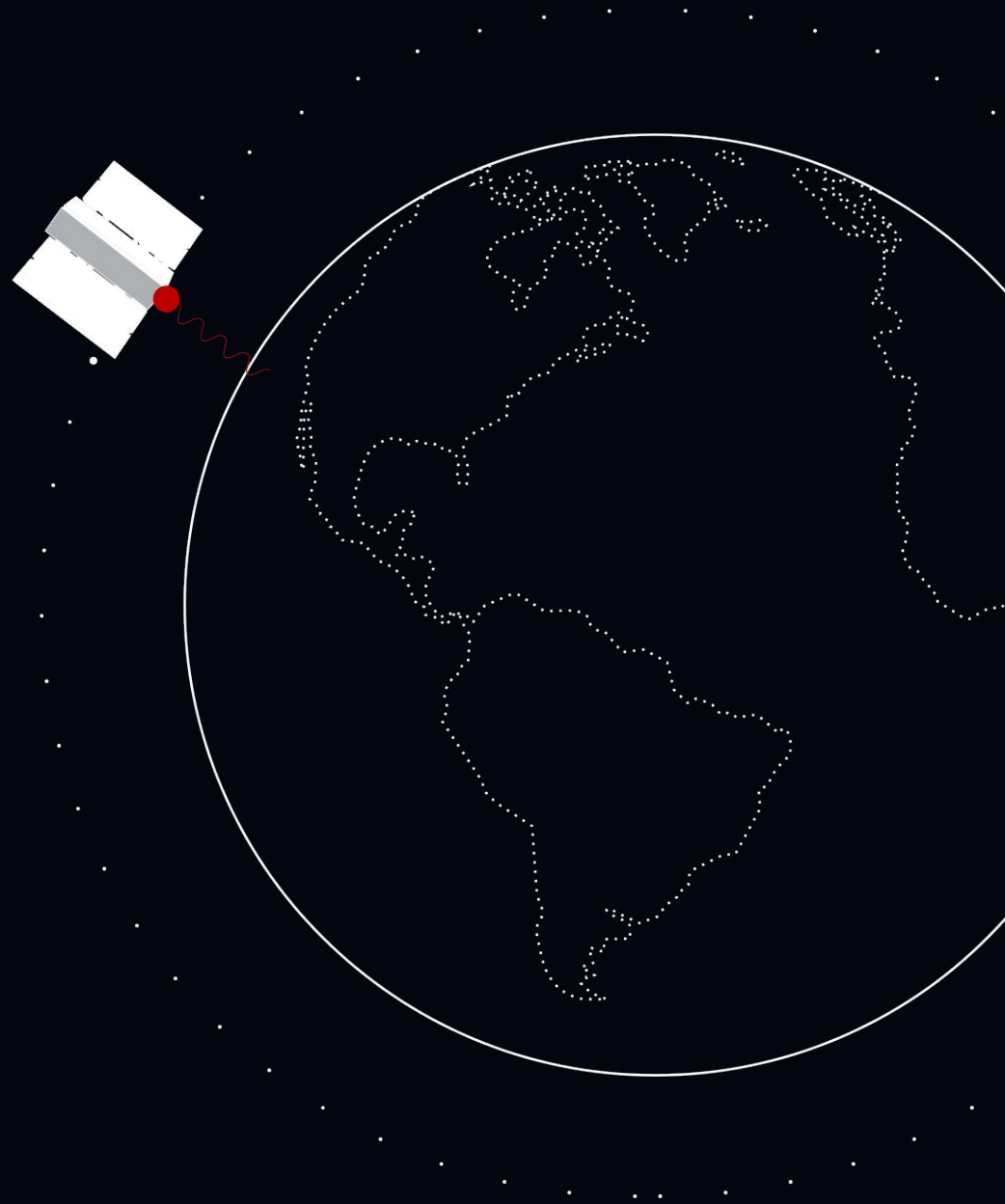
Spire User Services

Spire Space Services consists of 3 primary offerings, and the nature of the offering dictates the scope of user services provided:

- **Software in Space**: customer-deployed software on existing Spire satellites
- **Payload in Space**: customer payloads hosted on a Spire satellite bus
- **Solution in Space**: purpose-built, end-to-end solutions for customers



Conclusions



spire

Thank you!

From our team, to yours.

jeroen@spire.com