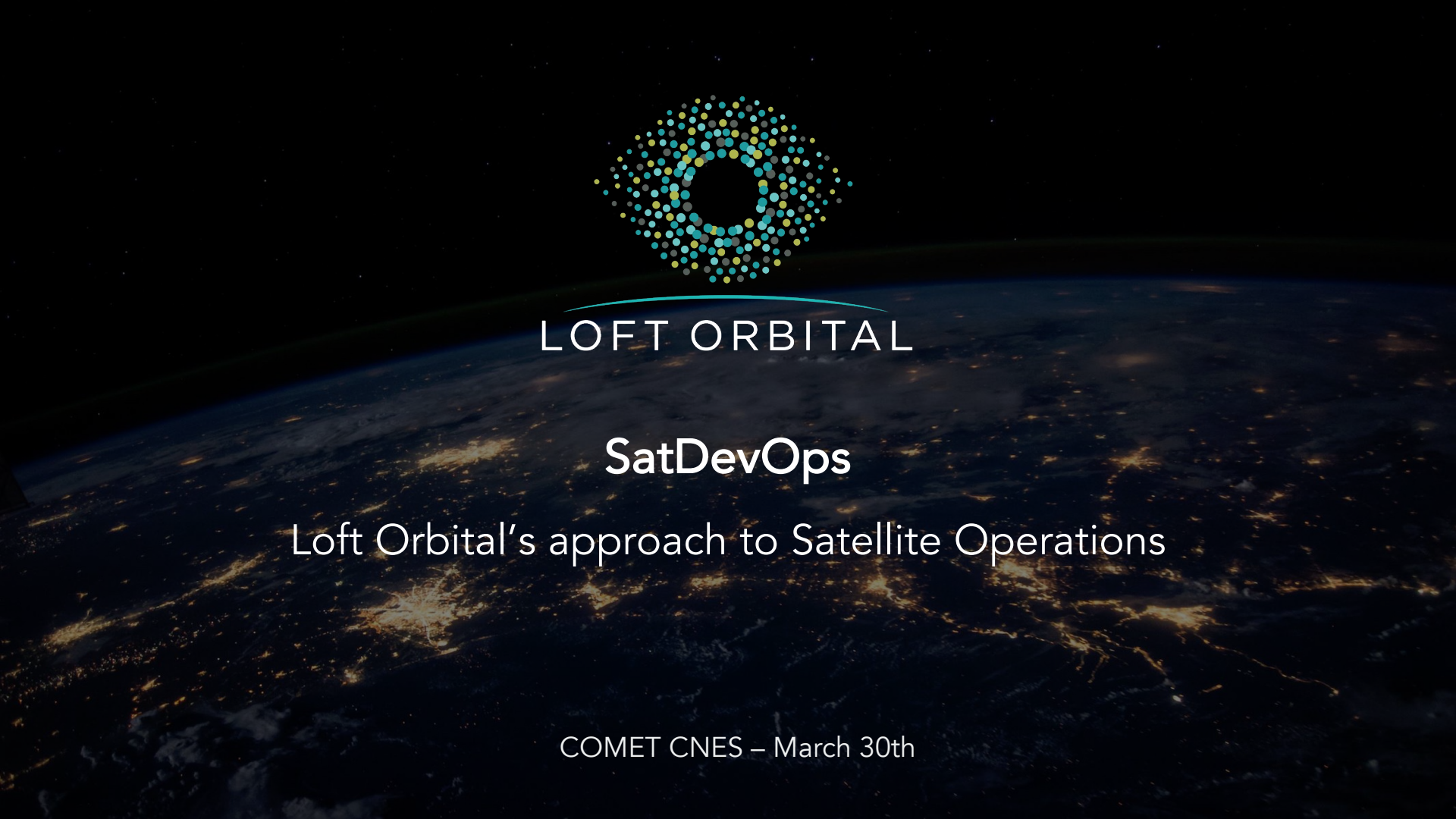



# SatDevOps

## Loft Orbital's approach to Satellite Operations

COMET CNES – March 30th

# What is SatDevOps?

**SatDevOps** is a *satellite operations methodology and philosophy* with which we can:

- operate satellites
- develop systems
- create a feedback loop between operations & development
- scale sustainably

to ***provide space infrastructure***.

There is no dedicated *satellite operations team* at Loft.

# Company

Toulouse, France

San Francisco, California

Golden, Colorado

FUNDING

**~ $200M**

EMPLOYEES

**> 135**

# Space Infrastructure

Loft is building a space infrastructure
to enable its customers to fly and operate any mission faster, simpler and safer.

## UNIVERSAL

*Any payload: physical or software-defined.*

*Support from one-off to mega-constellations.*

## FAST

*Accelerate schedule from design to orbit.*

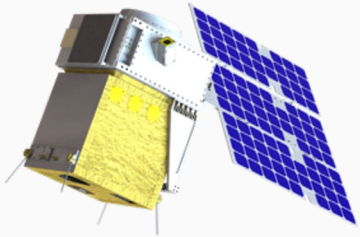
*Integrate within days, not months.*

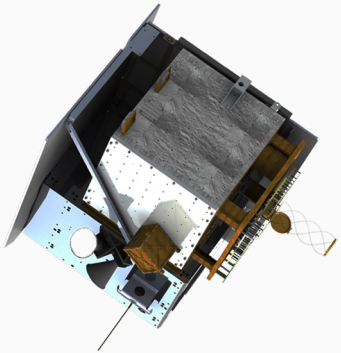
## SIMPLE

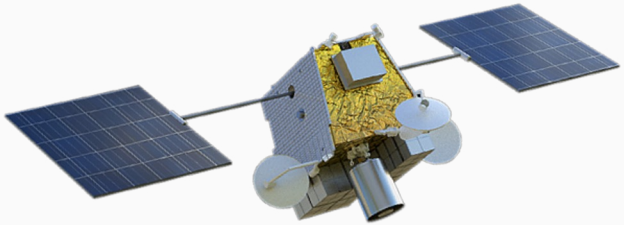
*No specific knowledge required to operate mission.*

*Full automation focusing on scalability and performance.*

# Platforms







# Customers

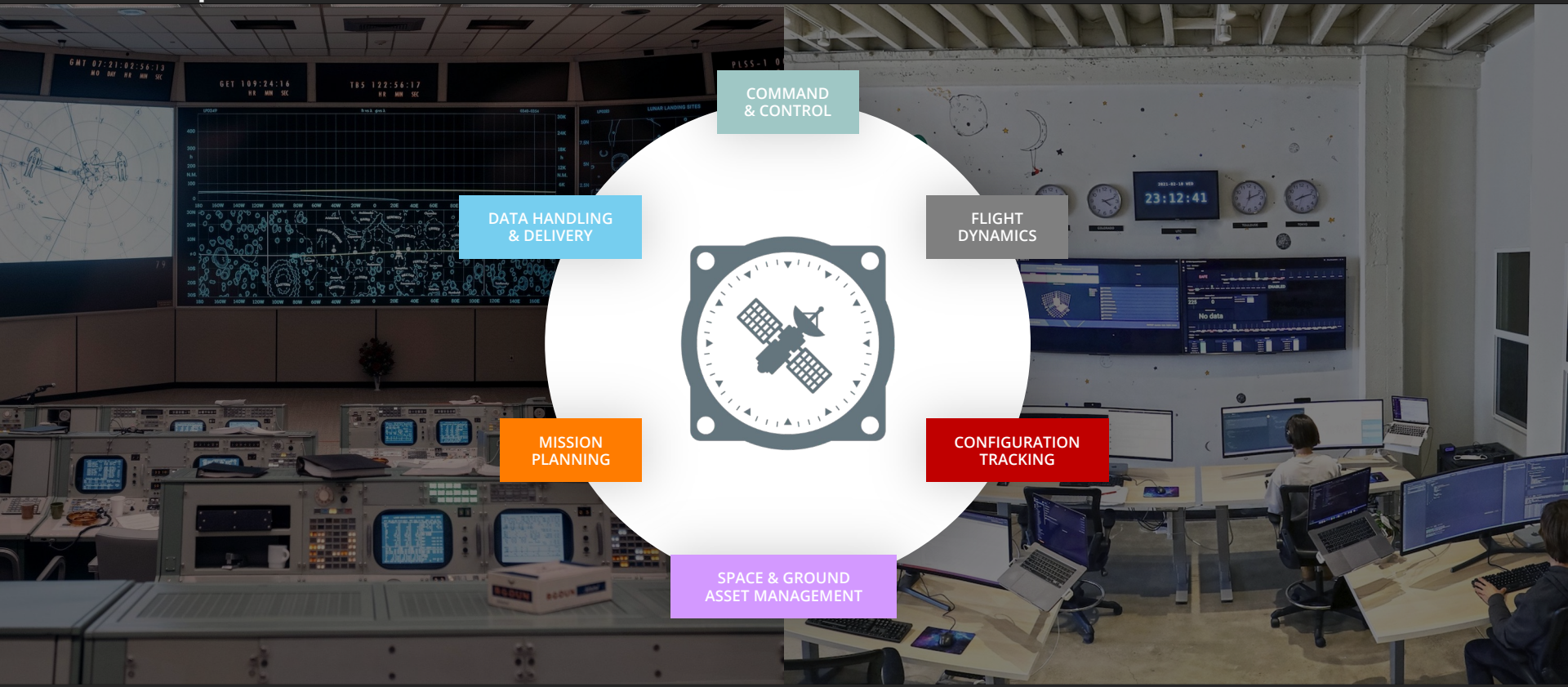
OUR CLIENTS

## INSTITUTIONAL & COMMERCIAL **CUSTOMERS TRUST** US

- **0 TO 70 KG** PAYLOAD CAPACITY
  > 70 KG IN DEDICATED CONFIGURATION
- **PHYSICAL AND MASSLESS** PAYLOADS
- **SUPPORTS** ANY APPLICATION IN LEO
- SINGLE **PAYLOADS** AND **CONSTELLATIONS**

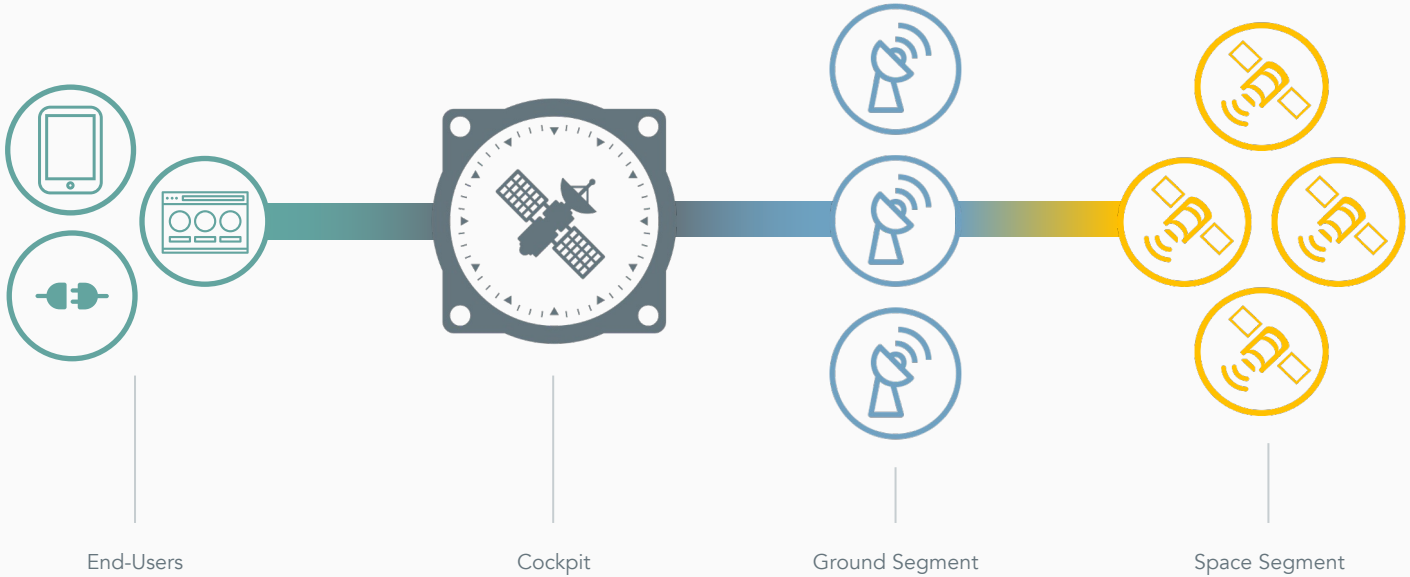
NASA · DARPA · eutelsat · Honeywell · CSA ASC · Microsoft · EarthDaily analytics · Ball Aerospace & Technologies Corp. · FUGRO · ONERA THE FRENCH AEROSPACE LAB · SCIENTIFIC SYSTEMS

# Cockpit Mission Control System



# Background: Space Infrastructure

Cockpit empowers end-users with powerful satellite constellation management capabilities while abstracting away the complexity of spacecraft operations.



# High levels of automation

**1**

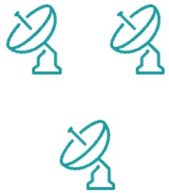
Customer tasks payload
*Via Cockpit's UI or API*

**2**

Cockpit schedules spacecraft commands

**3**

Schedule sent to satellite via ground station



**4**

Data is delivered to the client

# What is SatDevOps?

**SatDevOps** is a *satellite operations methodology and philosophy* with which we can:

- operate satellites
- develop systems
- create a feedback loop between operations & development
- scale sustainably

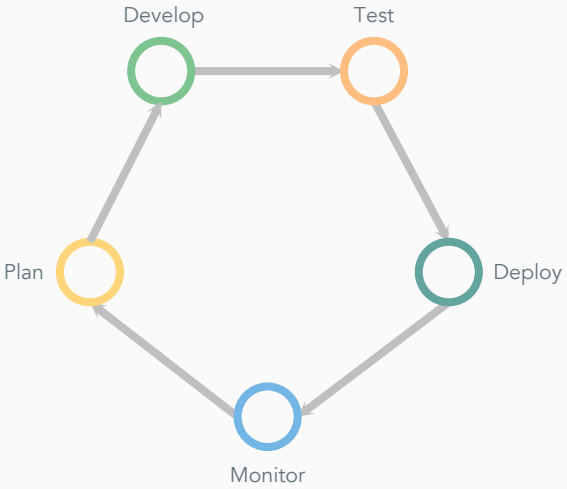
to ***provide space infrastructure***.

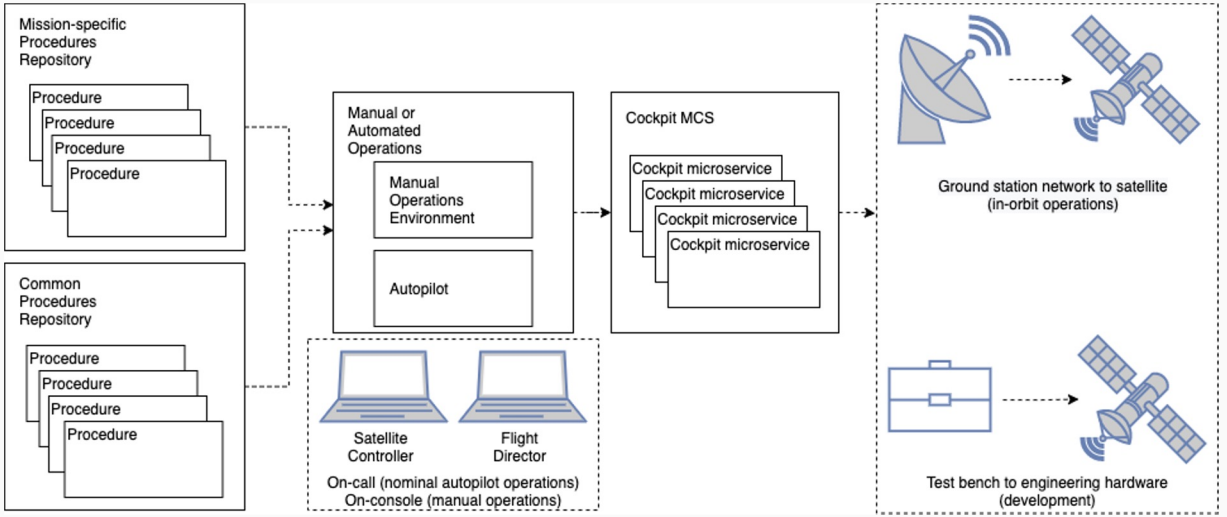
There is no dedicated *satellite operations team* at Loft.

# SatDevOps Core Competencies

1. Build processes, then automate them
2. Abstracted automation for a heterogeneous fleet
3. Infrastructure for on-call (not on-shift) operations
4. Test-as-you-fly
5. Unified interfaces usable by both software and people
6. Train developers and engineers as operators

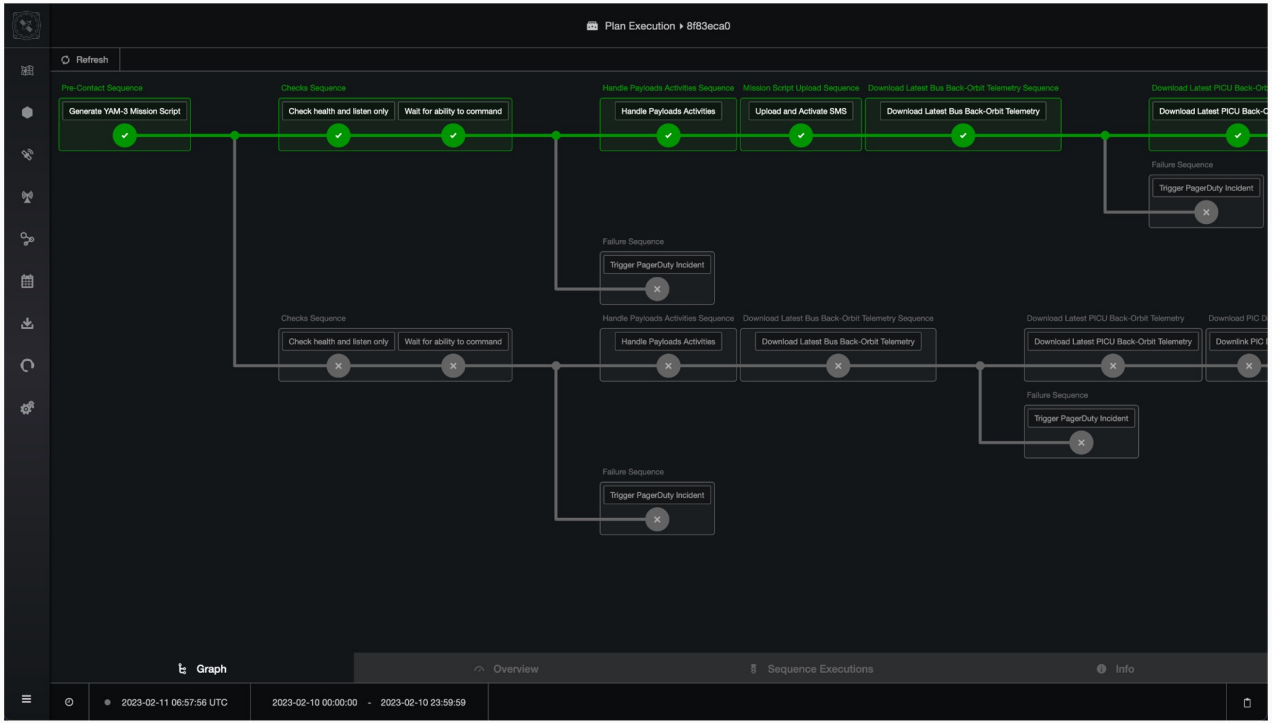
# 1. Build processes, then automate them




DevOps heritage; SatOps rigor: operations velocity.

# 2. Abstracted automation for space infrastructure



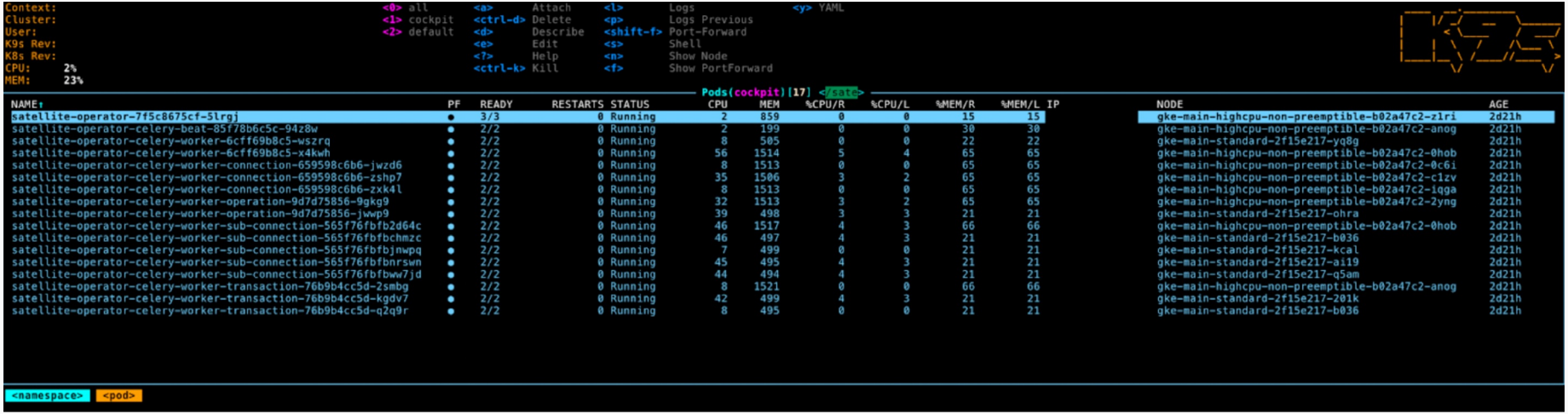

Reusability, standardization, modularity—simplify for both the operator and developer.

# 3. Infrastructure designed for on-call

On-call

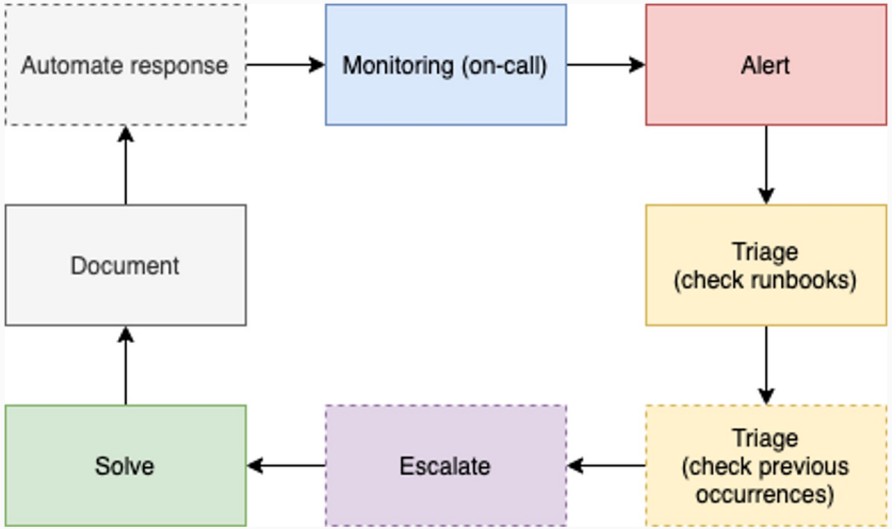
On-shift

# 3. Infrastructure designed for on-call




Software Infrastructure

# 3. Infrastructure designed for on-call




Process Infrastructure

# 3. Infrastructure designed for on-call

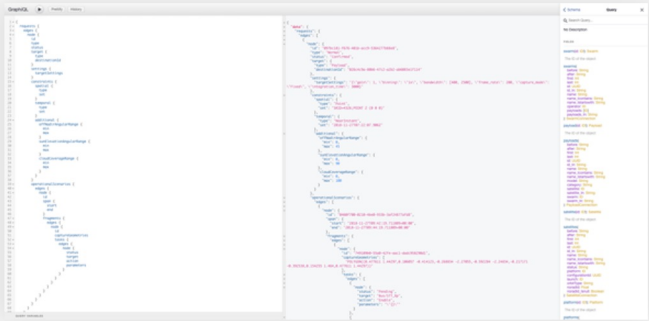

Human Infrastructure

# 4. Test-as-you-fly


Test/Fly: same systems, automated, operations-oriented.

# 5. Unified Interfaces

Leveraging the latest web technologies,
Cockpit exposes interfaces for interactive and programmatic access.



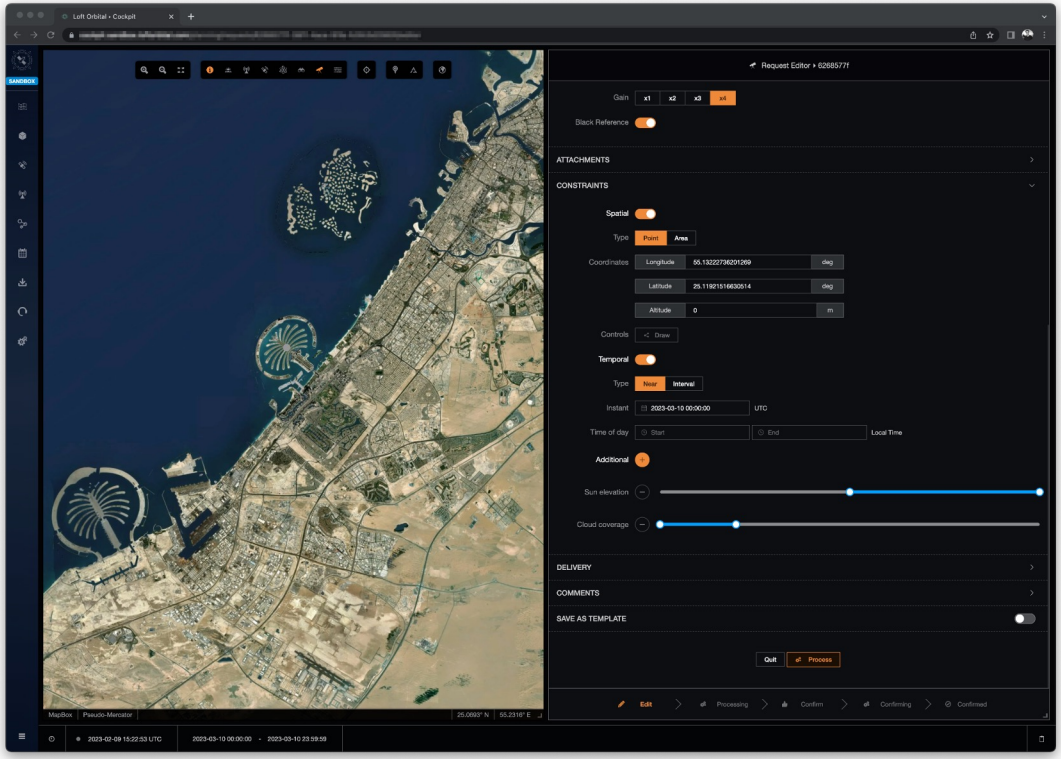
Graphical Web Apps

APIs

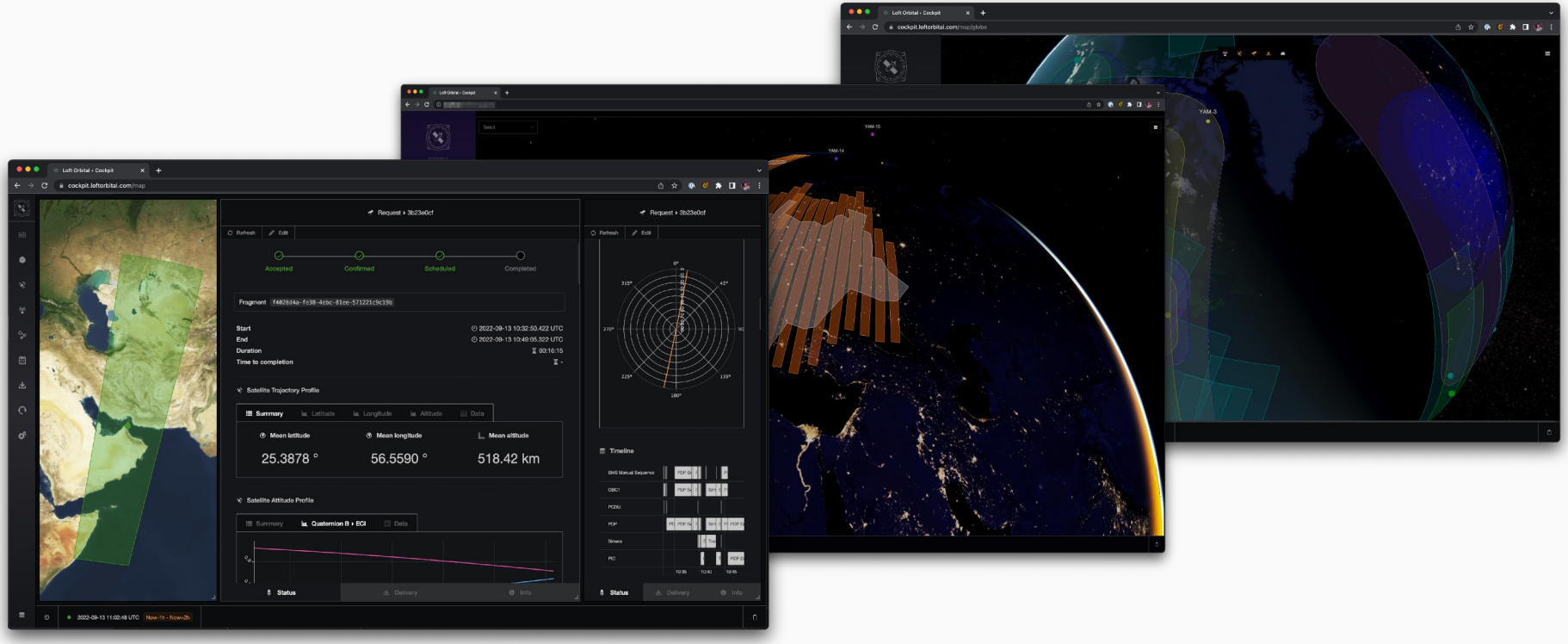
GraphQL

# 5. Unified Interfaces

```python
from cockpit.asset import Payload
from cockpit.mission import Request

# Create request
request = Request.create(
    target=Request.Target(
        type=Request.Target.Type.Payload,
        destination=Payload.get(name="My Thermal Infrared Imager")
    ),
    settings=Request.Settings(
        target_settings={"exposure_time": 1000, "gain": 4},
    ),
    constraints=Request.Constraints(
        spatial=Request.Constraints.Spatial(
            type=Request.Constraints.Spatial.Type.Point,
            set="SRID=4326;POINT Z (55.2962 25.2770 0.0)",
        ),
        temporal=Request.Constraints.Temporal(
            type=Request.Constraints.Temporal.Type.NearInstant,
            set="2023-03-10T00:00:00Z",
        ),
        additional=Request.Constraints.Additional(
            sun_elevation_angular_range=(0.0, 90.0),
            cloud_coverage_range=(0.0, 20.0),
        ),
    ),
    comments="Collecting thermal infrared data over Dubai.",
)

# Submit request for processing
await request.process()

# Select and confirm first available operational scenario
await request.confirm(request.operational_scenarios.first())
```



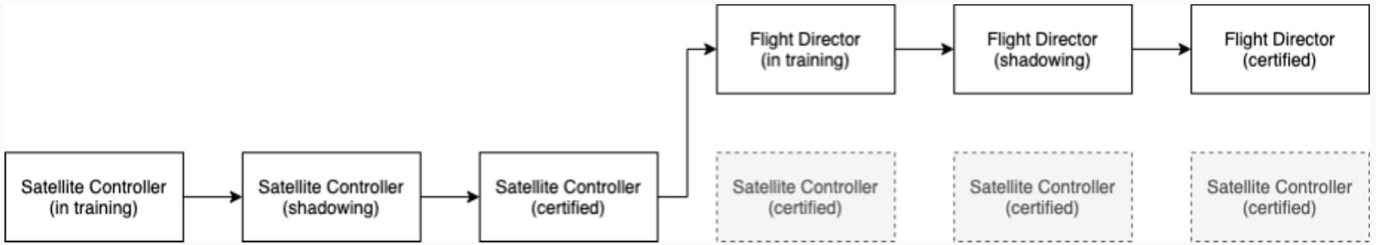
# 5. Unified Interfaces



# 6. Training

**SatDevOps Academy**

- SatDevOps Academy
- Course: Loft Orbital
- Course: Cockpit
- Course: Satellite Operations
  - OPS-1.1: What is Satellite Operations?
  - OPS-1.2: Spacecraft Overview
  - OPS-1.3: SatOps Tools and Infrastructure
  - OPS-1.4: Communication & Alerting
  - OPS-1.5: Procedures
  - OPS-1.6: SatCon Workflows and Responsibilities
  - [WIP] OPS-2.1: Flight Director Workflows and Responsi...
- Cohorts

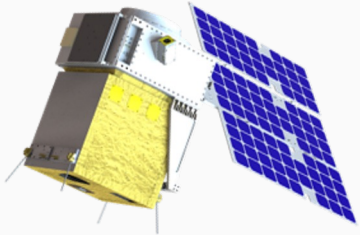


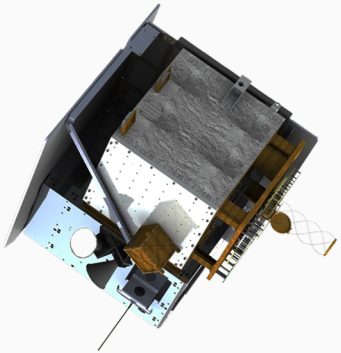

SatDevOps Academy

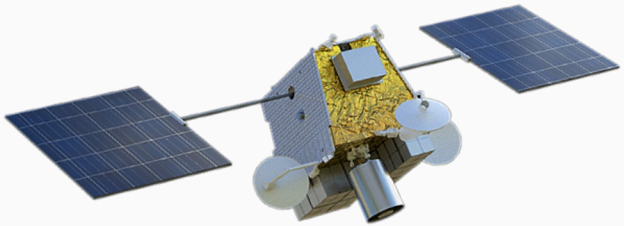
# Implications

- Space made simple—for everyone
- Efficiency, velocity, & quality
- Create better mutual understanding
- Build a good product—and build the *right* product

# Case Study: YAM-2, YAM-3, YAM-5, Beyond







- Monday to Friday: SatCon to implementation
- The *wisdom of production*
- Beyond operations

LOFT ORBITAL

# THANK YOU

Space Made Simple.