MISSION ASSURANCE FOR LARGE CONSTELLATIONS OPERATIONS

From Risk to Resilience: Evolving Mission Assurance for Mega-Constellations

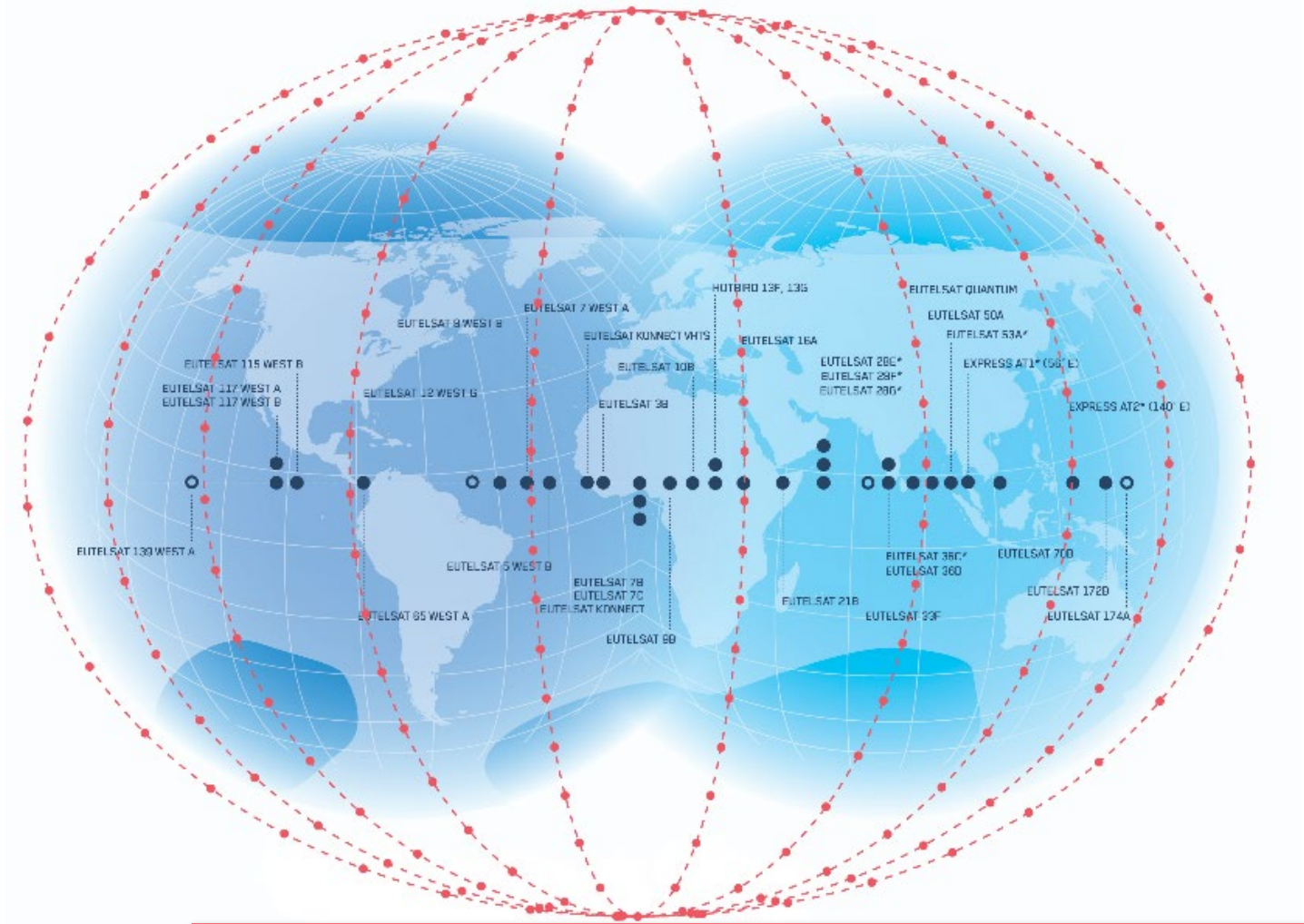
Dan Kroboth
dan.kroboth@eutelsat.net

January 21, 2026

Introduction

- Dan Kroboth – dan.kroboth@eutelsat.net
- Vice President, Satellite Operations for Eutelsat
- Eutelsat-OneWeb is name of our LEO Connectivity Business
  - 650+ OneWeb GEN1 satellites in Low Earth Orbit (LEO)
  - The LEO satellite operations team is responsible for ensuring the satellites are available to deliver the OneWeb network to customers around the world.
- Began career in August 1999 as a software developer for TRW (US Aerospace Company, now Northrop Grumman)
- First part of my career was focus on software development for large scale systems integration projects (Japanese Defense Logistics, Financial Crimes Reporting, Missile Defense Data Integration)
- After 9 years, joined a satellite startup, designed, implemented, and ready for launch a $20M LEO satellite in 2 years.
  - Designed the Avionics; Wrote the Flight Software; Designed the Space-Ground Interface; Trained the Operations Teams
- Joined OneWeb in August 2015 as part of operations design team; moved to Toulouse France; Back to Washington, DC to support Operations
- Operated satellites for small startup (fully lights out), US Government, and commercial

First Commercial LEO & GEO Operator



Operating 650+ LEO satellites with rapid deployment timelines and evolving system design reshapes traditional mission assurance.

GEO fleet and LEO constellation in 2026

34 GEO

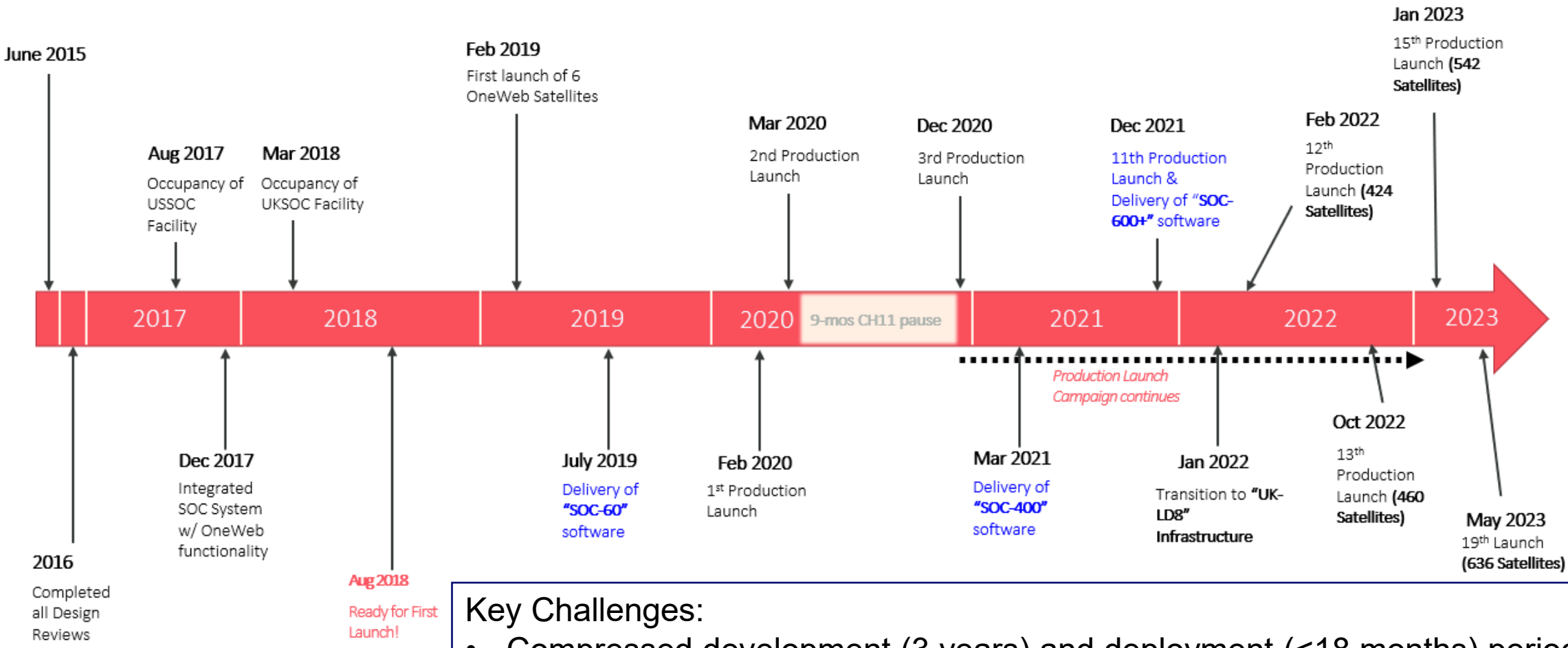
654 LEO in 12 Planes

~130 people in four control centers
~70 LEO & ~60 GEO

For the LEO Constellation in every day:

- ~1.78 satellite years
- ~10,000 contacts
- ~50,000 procedure executed
  - 97.7% procedure success rate
  - ~64 days of procedure runtime
- ~540,000,000 Telemetry Packets
- ~27,000 Conjunction Data Message
- ~50,000 commands / satellite / day
  - ~40K Service, 5K Ground, 2K FDIR

Concepts for OneWeb Satellite Operations



Key Challenges:

- Compressed development (3 years) and deployment (<18 months) periods
- Satellite, ground, and network systems maturing in parallel
- Need to scale operations from dozens to 1,000+ spacecraft
- High-reliability service expectations across a LEO network

eutelsat

How Do We Make Mission Operations Work

Adaptive Frameworks: Flexibility, Openness, Scalability

Flexible Architectures
- Non-interrupting upgrades as constellation grows
- End-to-end design to handle evolving requirements
- Global mission visualization for situational awareness

Open Systems Approach
- Mix of COTS + custom software
- Avoid vendor lock-in
- Modular software for rapid iteration and improvements

Scalable Infrastructure
- Cloud-integrated SOC spanning UK/US
- Automation ramps as constellation expands
- Continuous migration to AWS to reduce non-value-added work (70% reduction goal)

Results: Highly Automated Operations Systems Requiring Minimal Oversite during non-change periods.

Typical night/weekend operations shift
- 1 Mission Director
- 1 Operation Engineers

On Stand-by
- 1 Flight Dynamics Engineer
- 1 Operations Engineer

Objective: Deliver 99.95% Communications Mission Availability at the satellite level.
(4.4 hours of downtime per satellite per year ~ average human involved recovery 12h)

Automation as a Mission Assurance Multiplier

Challenges:

The satellites are spread across 12 planes. However, each plane is a different Solar Beta angle yielding:

- Different power available & thermal environment
- See a given customer at a different time of day
- Therefore, every plane has a different plan

Each satellite is “slightly” different

- Differences in calibration / ageing / performance
- Some satellites are very different due to hardware failures
- Orbit drift means each satellite executes a slightly different plan

Why Automation Is Essential:

- Human-centered ops don’t scale to hundreds of satellites
- “Lights out” operations reduce operational risk and human error
  - Human intervention directed towards problem resolution
- Automated prevention → earlier anomaly detection → higher availability

Examples from OneWeb:

- Automated Flight Dynamics: orbital state, COLA, maneuver planning
- Event-driven alerts & automated remediation
- Command-and-control microservices to improve performance and reduce compute load

Automation ensures that once tested, the system executes the same pattern.

OneWeb Satellite Operations Perimeter

- ~110 Microservices spread across 4 major elements (Command & Control, Flight Dynamics, Planning, Orchestration)
- ~900 satellite procedures + ~200 ground / utilities procedures, ~70 in-common usage
- ~280 Quick Reaction Handbooks
- On-Board Satellite Automation

Inputs:

- 3 * 600 (32 Hour Span) Communications Plans
- Conjunction Data Messages
- External Equipment Availability (TTC Stations)

Outputs:

- Satellite Ephemeris (including planned maneuvers) and Attitude for Service Planning and Conjunction Screening
- TTC Contact Plan

For the LEO Constellation in every day:

- ~1.78 satellite years
- ~10,000 contacts
- ~50,000 procedure executed
  - 97.7% procedure success rate
  - ~64 days of procedure runtime
- ~540,000,000 Telemetry Packets
- ~27,000 Conjunction Data Message
- ~50,000 commands / satellite / day
  - ~40K Service, 5K Ground, 2K FDIR

- Each satellite is “different” executes a slightly different plan

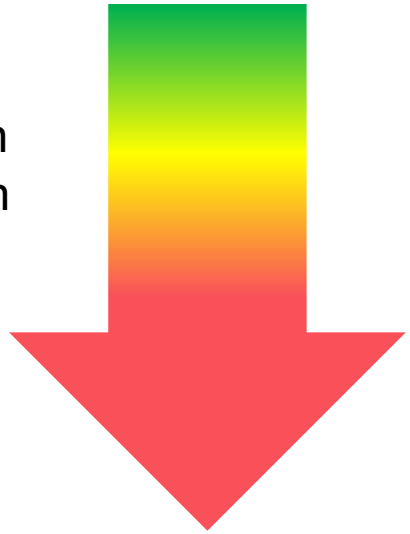
At our operating scale, it is not realistic to test all combinations / contingencies

Hierarchy of Pain & Philosophy of Avoidance

Acceptable

Acceptable with prior notification

Unacceptable



- Problem Impacting Non-Service Satellite
- Problem Impacting Single Service Satellite
- Problem Impacting Service Satellite/Plane
- Problem Impacting Entire Service Plane
- Problem Impacting Entire Service Fleet
- Problem Impacting Entire Satellite Fleet

Fail Early, Fail Fast, Fail Recoverable, Fail Limited

Enabling Environments & Resources

- Element Development Environment
  - Dedicated per element (Command and Control, Flight Dynamics)
  - Access to Communications Plans, Satellite And Ground Telemetry allowing for test against the “real data”
  - Integrated testing of code and services with the environment
- Integration Environment / Disaster Recovery Environment
  - Cloud Infrastructure implemented as code.
  - Can deploy a duplicate of our production environment as a test resource to do integrated validation of a new major version
- Production Environment (Live Environment)
  - Micro-services architecture allows in-situ substitution or parallel versions
  - Environment Variable Tuning by satellite allows targeting to specific version API/procedure/etc to a satellite or set of satellites

Test Assets:

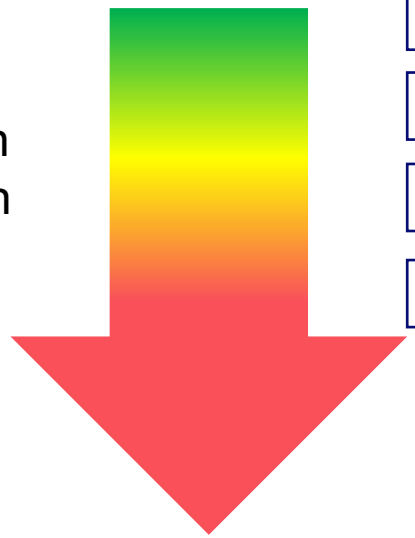
- Space Vehicle Simulator
  - Runs as on virtual machine
  - Simulation of the Flight Computer, emulation of other spacecraft hardware
  - Executes Real Flight Software Image
  - Ground to Space Interface Compliant
  - Can be configured in any orbit / failure condition
- Flatsats
  - Physical Flight Computer + Platform Avionics
  - Execution on Real Hardware of Procedures
  - Executes Real Flight Software Image
  - Ground to Space Interface Compliant
- Non-Service Satellites (~50)
  - Non-Service Capable Vehicles and Tech Demonstrators, same platform different payload
    - Not in mission orbits
  - Spare Satellites / Calibration Fleet

Rollout Philosophy

Acceptable

Acceptable with prior notification

Unacceptable



Prerequisite Action Performed		Problem Discovery Avoid
Test in Development Environment		
Test against SVS		
Test against FlatSat		
Deploy on “Non-Service Capable” Satellite	→	Problem Impacting Non-Service Satellite
Deploy on all “Spare/Calibration” Satellites	→	Problem Impacting Single Service Satellite
Deploy on “Service” Satellite	→	Problem Impacting Service Satellite/Plane
Deploy on one “Service” Satellite per Plane	→	Problem Impacting Entire Service Plane
Deploy on one Plane	→	Problem Impacting Entire Service Fleet
		Problem Impacting Entire Satellite Fleet

Fail Early, Fail Fast, Fail Recoverable, Fail Limited

Mission Assurance Philosophy

- Assure through automation
- Mission assurance from patterns, not per-satellite checks
- Contain, not eliminate, failures
- Resource-based prioritization

Mission assurance at constellation scale requires: automation, flexible frameworks, adaptive operations, and a culture that embraces controlled failure

Questions?