# The Sentinels Mission Control Systems – a versatile approach to deployment and operations

Michael Koller

01/03/2016

Issue/Revision: 1.2
Reference:
Status: Approved
ESA UNCLASSIFIED - Releasable to the Public



European Space Agency

# Agenda

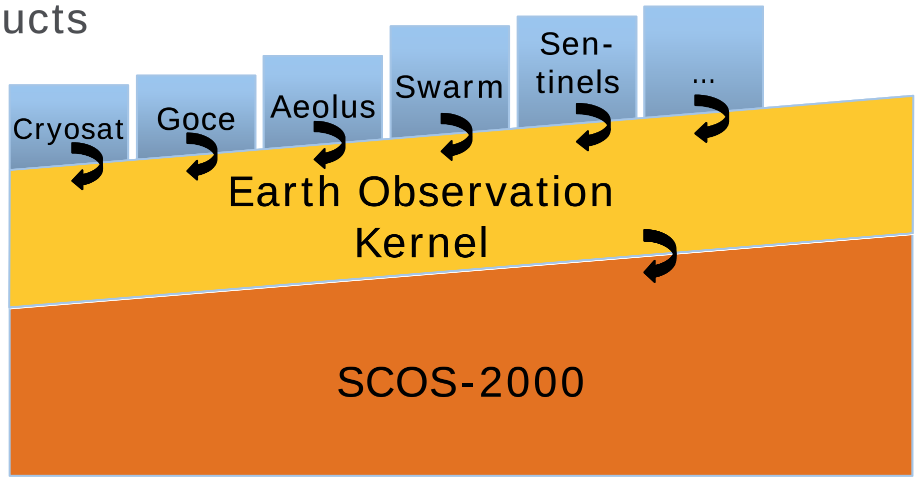
- Background
    - ESA's Mission Data System Kernels
    - Current lifecycle of a Mission Control System
    - Copernicus fleet of spacecraft

- Dealing with change – a versatile approach to system maintenance

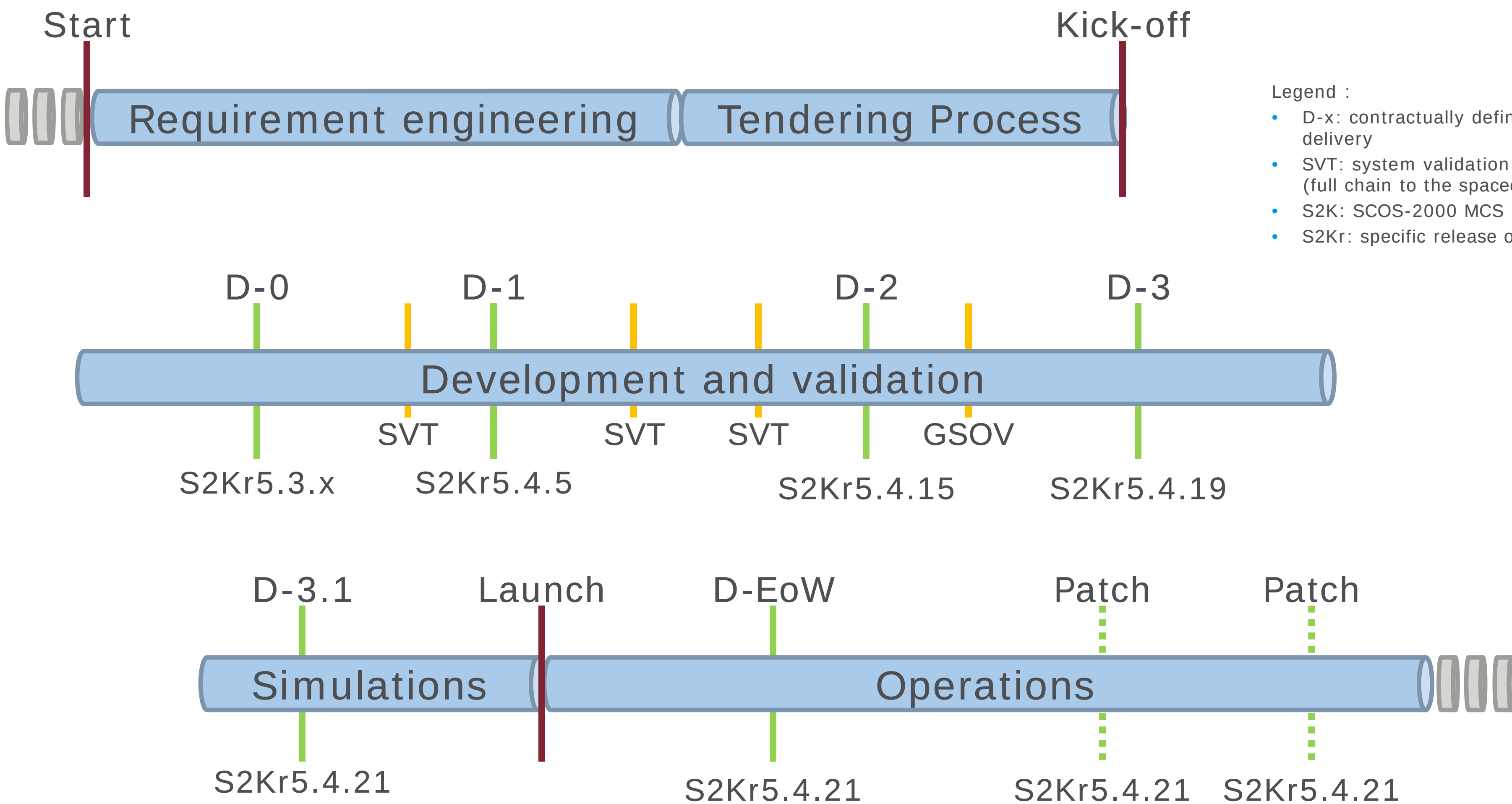
- Conclusion

# ESA's software kernel(s)

- Common system kernels for Mission Data Systems
  - Integrated by all missions
  - Mission Control System: SCOS-2000
  - Operational Simulator: Simulus Toolset
- Basic functionality shared between all missions
- Further enhanced by mission family kernels
- Highly configurable software products
- Living systems
  - Growing
  - Evolving



# Current life of a Mission Control System (MCS)

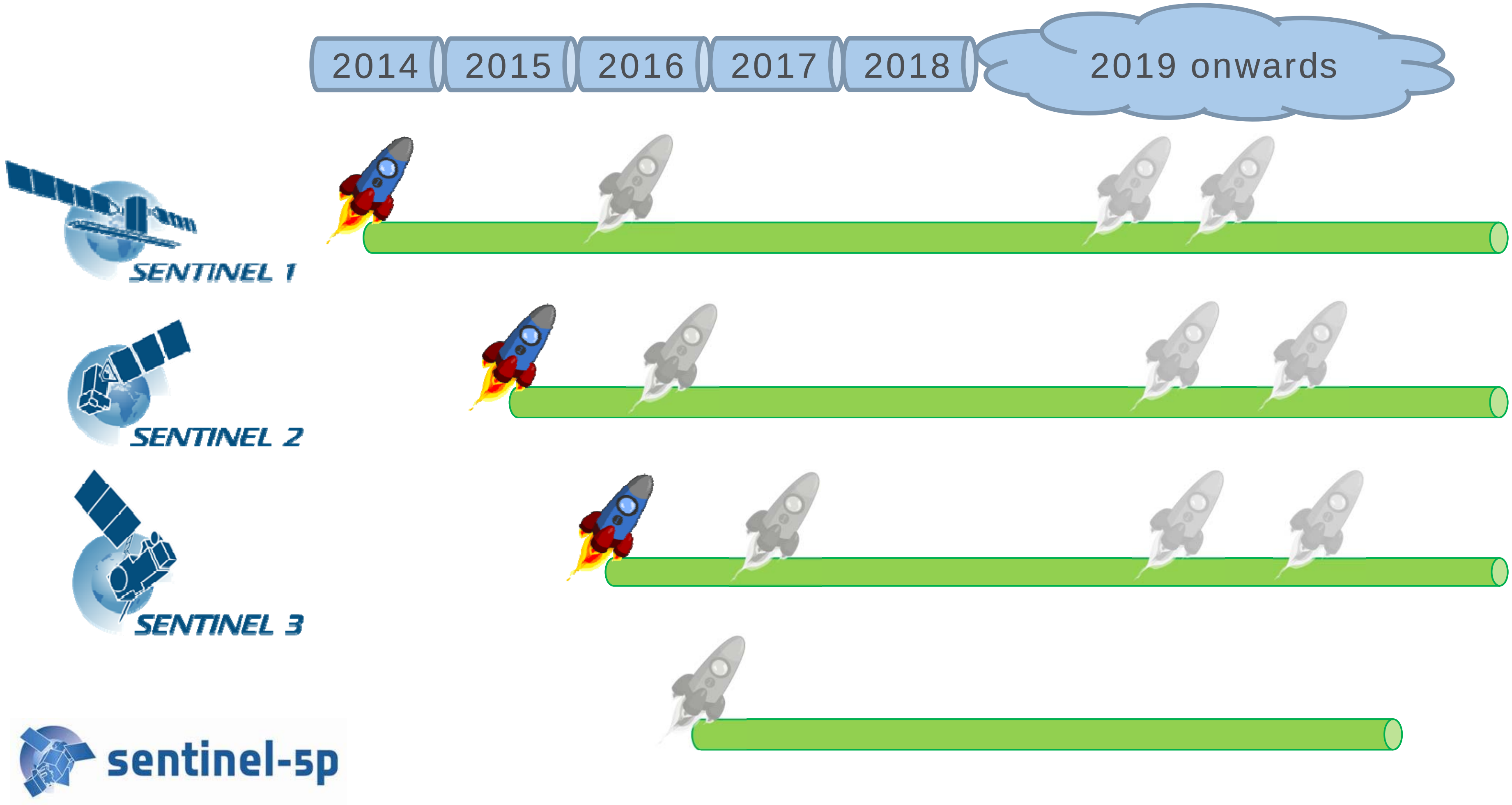



Legend :

- D-x: contractually defined delivery
- SVT: system validation test (full chain to the spacecraft)
- S2K: SCOS-2000 MCS kernel
- S2Kr: specific release of S2K

Not all details shown (reviews, etc)

# Copernicus Fleet of Spacecraft




Grayed-out rockets indicate a launch date in the future.

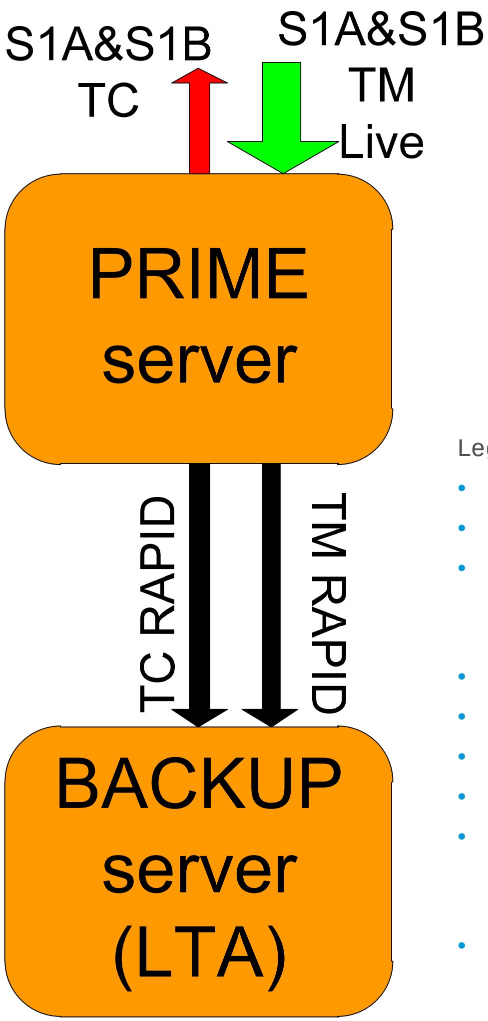
# Drivers for the Sentinels MCS

1. Long mission lifetime
   a. Requires updates to kernel, soft- and hardware baselines
   b. Long term evolution plan
2. Staged deployment of spacecraft
   a. Changing configuration of the system
   b. System validation and preparation without interference
   c. Specific requirements for different mission phases

- Resulting Challenges
  - Limited validation resources due to parallel operations
  - Introduction of regression issues by system updates
  - Difficult migration of system archives
  - Interference with flying spacecraft
  - Avoid falling behind
  - Budgetary constraints

# Dealing with different mission phases - Routine Setup

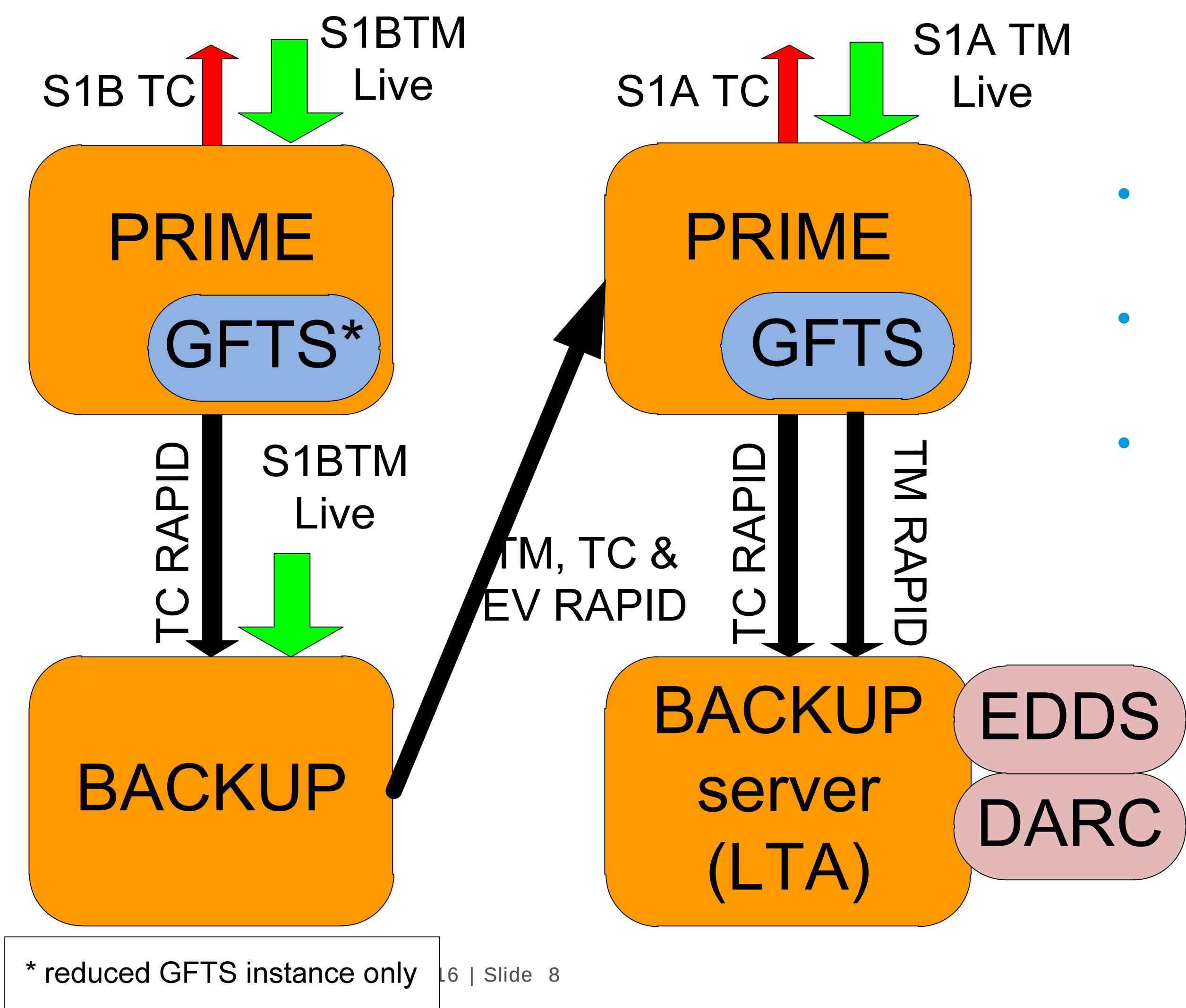
- 100% availability vs reconfigurations and flexibility in preparation phase
- High number of passes and workstations during LEOP and resulting system load
- (simulation of) Contingency scenarios
- Risk of operator errors (working on the wrong Spacecraft)




Legend (also for next slide):

- TM: telemetry
- TC: telecommand
- RADID: file format to exchange spacecraft data between different system instances
- S1A: Sentinel-1a model
- S1B: Sentinel-1b model
- LTA: long term archive
- GFTS: generic file transfer system
- EDDS: EGOS Data Dissemination System (portal for external users to access packet archive)
- DARC: data archive (parameter archive accessible to external users)

# Dealing with different mission phases – LEOP Setup




- Dedicated hardware shared across missions
- Forwarding mission data for routine operations
- Reusing external interfaces to reduce end-to-end validation

# Dealing with constant evolution of baselines

- Keeping up with new baselines
    - Don't rush – keep number of full deployments reasonable
    - Don't be the first – wait until validated by other missions
    - Don't overwhelm – reduce big jumps to smaller hops
- Priorities and risk taking
    - Mission in hot phase gets most focus
        - Drives the priorities for fixes
        - First to receive support in case of issues
    - Mission with most time and non-critical activities takes new releases first
        - Benefit from commonalities – shared validation effort
        - Validate on non-flying mission first (e.g. SVT)

# Example for new baseline integration

Scenario: deploy “D-3.8” (including a new S2K baseline)

- Mission status
    - S-1a: in flight
    - S-1b: in preparation, last SVT completed
    - S-2a: in flight
    - S-2b: in preparation, next SVT in 5 weeks
    - S-3a: in simulation campaign, launch -45 days
- Deployment timeline
    1. Validation environment for all missions
    2. Dedicated environment for S-2b (and S-1b)
    3. Operational environment for S-3a
    4. Pause – gain confidence
    5. Operational environment for S-1a and S-2a

# Conclusion

## Sentinels MCS has to embrace change

- Long mission lifetime
  - Integrate new baselines (, but …)
  - Don’t rush
  - Don’t be the first
  - Don’t overwhelm
- Staged deployment of spacecraft
  - Dedicated (and shared) environment for preparation and validation