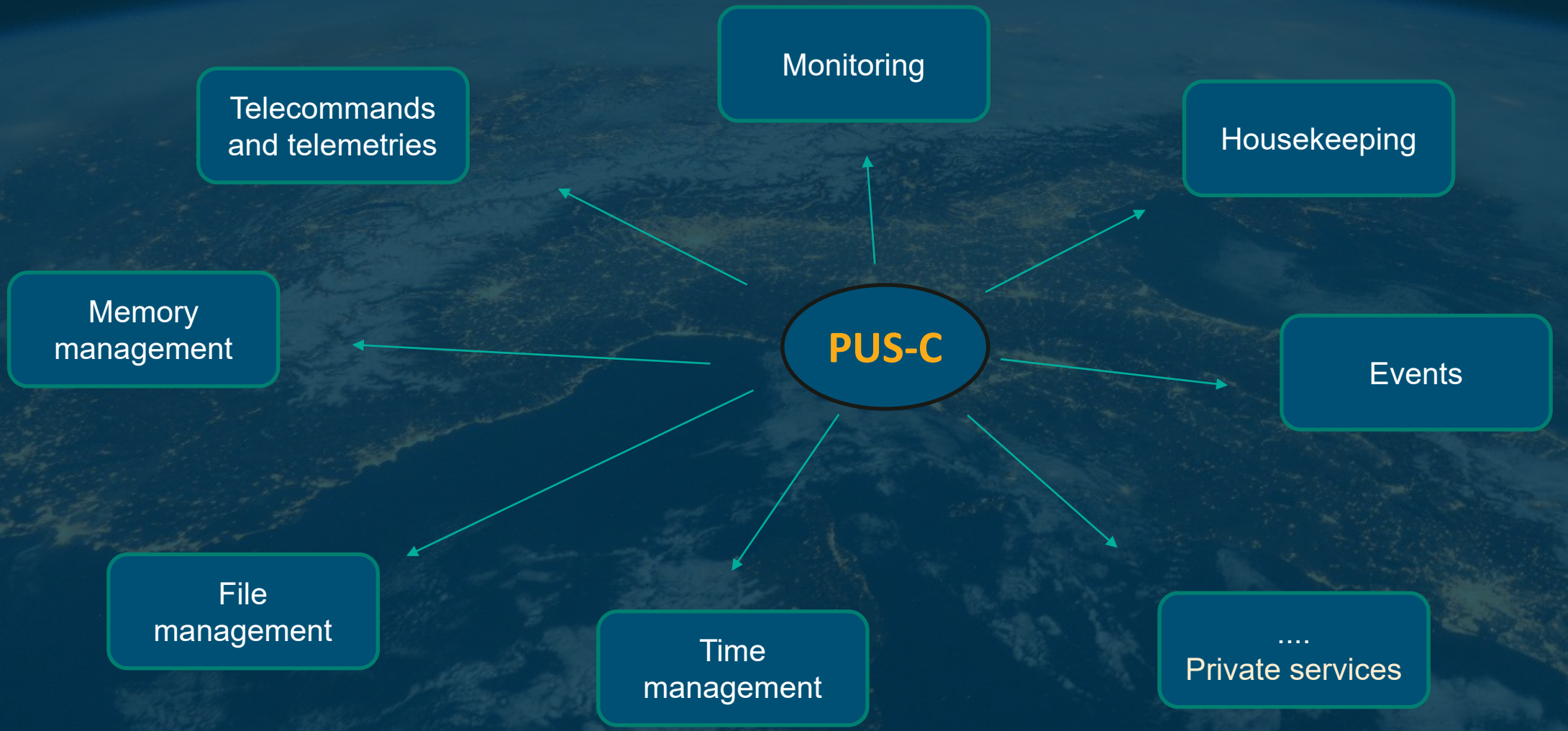
# Towards a PUS component-based execution platform

The PUS standard specifies on-board services for ground stations to operate the spacecraft. It is big and complex.



# The challenge


**PUS**
standard

Mission
requirements

**PUS**
core library

**PUS**
private services

**OBSW**

- System manager
- AOCS
- Unit managers
- Power & Thermal
- ...

*PUS Tailoring*

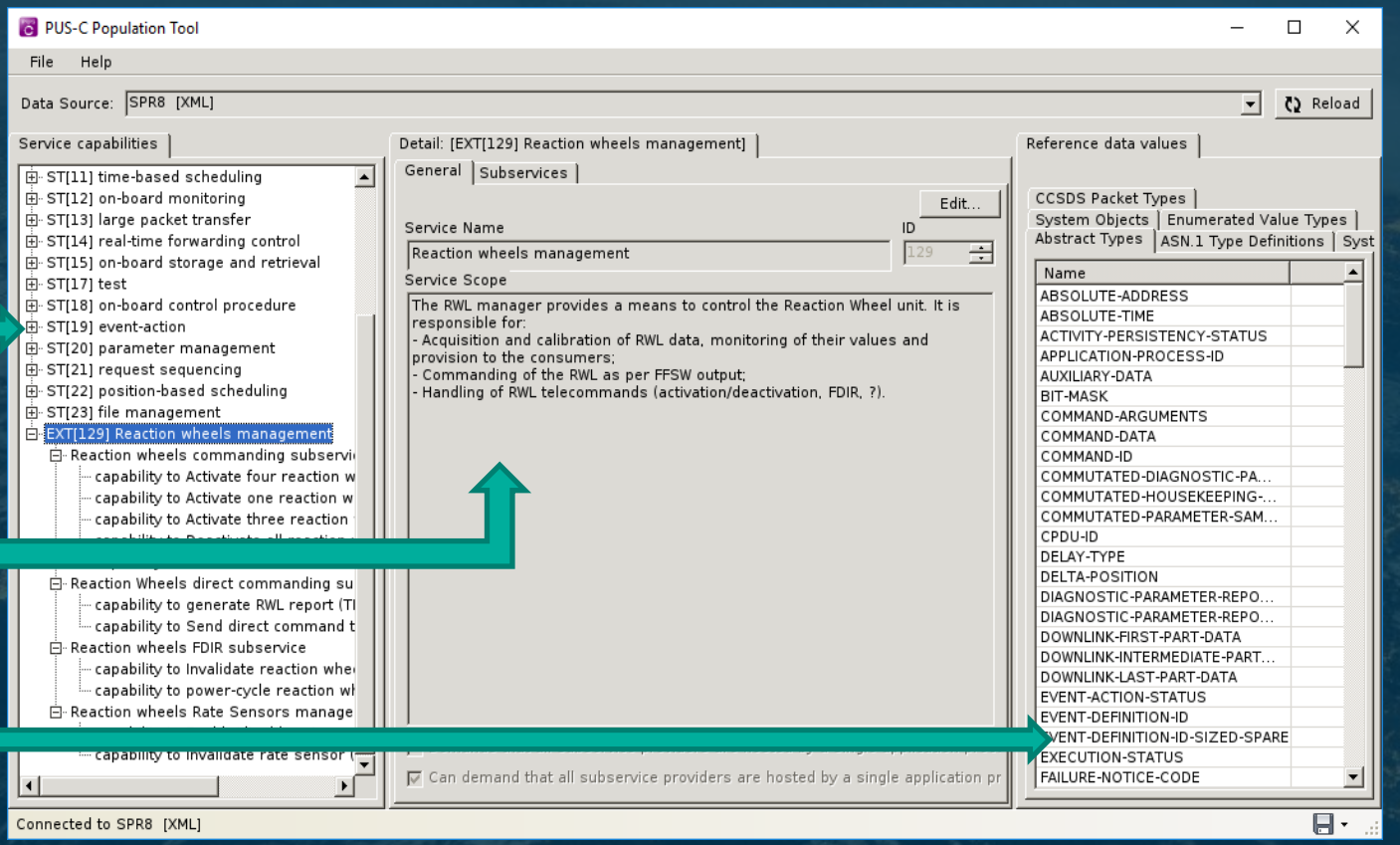
- Service subset
- APID
- Failure codes
- Memory areas
- Parameters
- Events
- ...

# opus: the PUS specification and tailoring tool

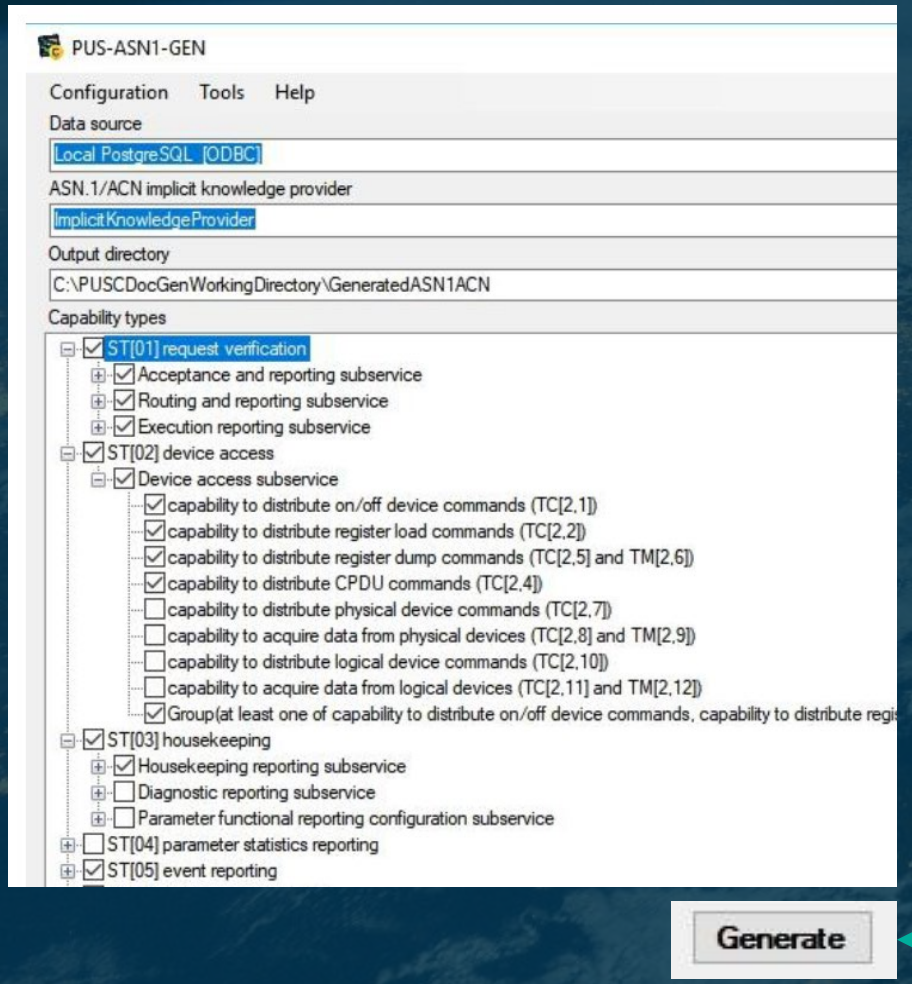
List of available PUS services and corresponding TM/TC packets

Service detailed specification

Project-specific configurable data



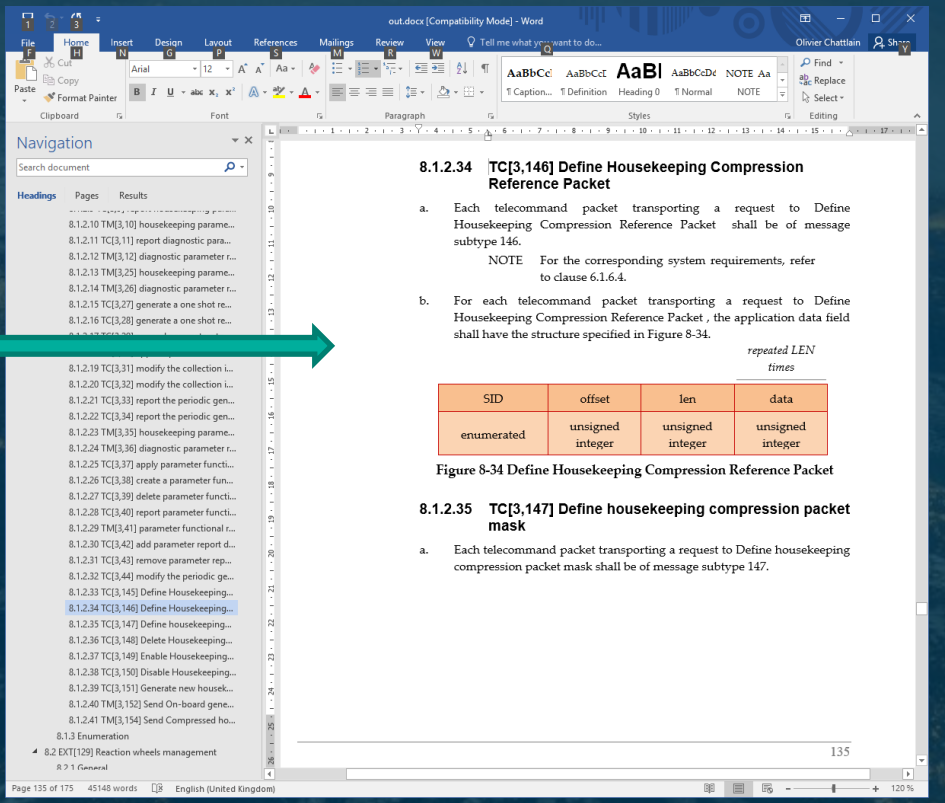
# opus tailoring process




Pick the PUS services you need, give your project specific data (APID, Failure codes…), and press the button

# opus document generation

The tools generate documents with a layout that is identical to the original PUS-C standard



# opus to TASTE models (TM/TC definitions in ASN.1)

**Telecommand** (SEQUENCE) ASN.1 ACN — Min: 14 bytes Max: 60 bytes

Finally instantiate the Telecommand type

| No | Field | Comment | Present | Type | Constraint | Min Bits | Max Bits |
|---|---|---|---|---|---|---|---|
| 1 | packet-version | | always | NULL | N.A. | 3 | 3 |
| 2 | packet-type | | always | NULL | N.A. | 1 | 1 |
| 3 | secondary-hdr | | always | NULL | N.A. | 1 | 1 |
| 4 | dest-apid | Possible values:<br>▪ *ground* (2047)<br>▪ *flight* (2046) | always | APID | N.A. | 11 | 11 |
| 5 | sequence-flags | | always | NULL | N.A. | 2 | 2 |
| 6 | packet-seq-count | | always | SEQ-COUNT-OR-NAME | N.A. | 14 | 14 |
| 7 | packet-data-len | | always | NULL | N.A. | 16 | 16 |
| 8 | secondary-header | | always | Telecommand-secondary-header | N.A. | 35 | 35 |
| 9 | packet-data | | always | TCs | N.A. | 22 | 397 |

**TCs** (CHOICE) ASN.1 ACN — Min: 3 bytes Max: 50 bytes

Create the full list of TCs used in my project

| No | ACN Parameters [?] | Type |
|---|---|---|
| 1 | tc-type | TYPE-ID |
| 2 | tc-subtype | TYPE-ID |

| No | Field | Comment | Present | Type | Constraint | Min Bits | Max Bits |
|---|---|---|---|---|---|---|---|
| 1 | tc2-4 | | tc-type=2 AND tc-subtype=4 | TC-2-4 | N.A. | 22 | 397 |

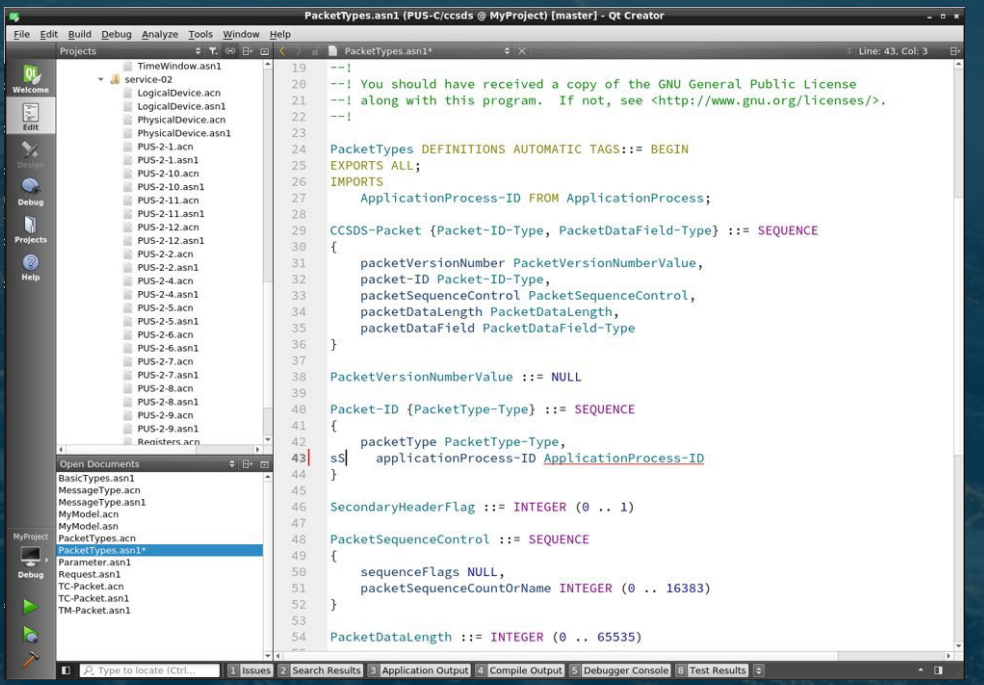
**TC-2-4** (SEQUENCE-OF) ASN.1 ACN — Min: 3 bytes Max: 50 bytes

Instantiate the TC-2-4 for my project:

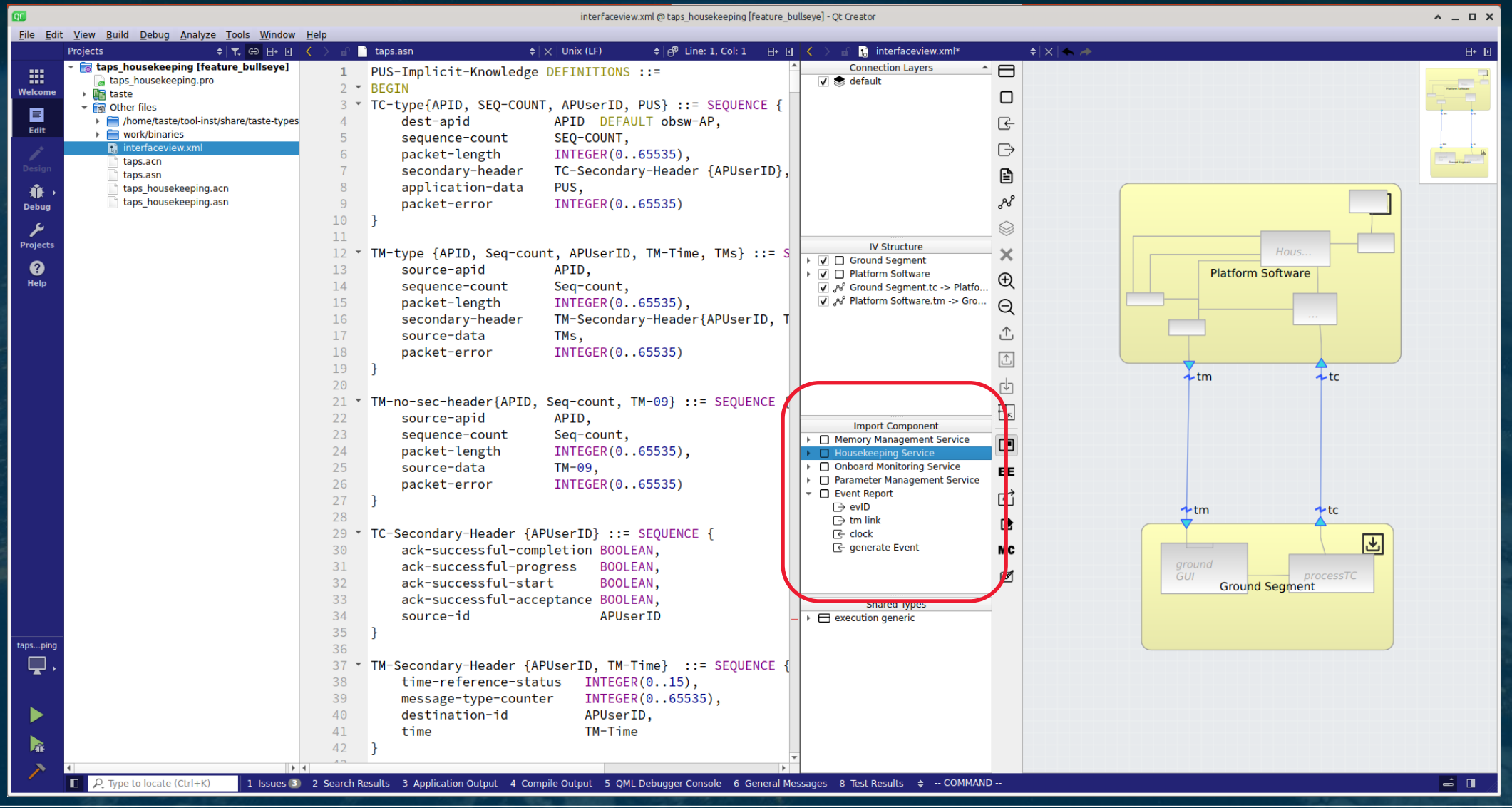
| No | Field | Comment | Type | Constraint | Min Length (bits) | Max Length (bits) |
|---|---|---|---|---|---|---|
| 1 | Length | Special field used by ACN indicating the number of items. | unsigned int | (SIZE(1 .. max-Cpdu-Cmds)) | 1 | 1 |
| 2 | Item #1 | | CPDU-Cmd | N.A. | 21 | 198 |
| | ... | | | | | |
| 3 | Item #2 | | CPDU-Cmd | N.A. | 21 | 198 |

**CPDU-Cmd** (SEQUENCE) ASN.1 — Min: 3 bytes Max: 25 bytes

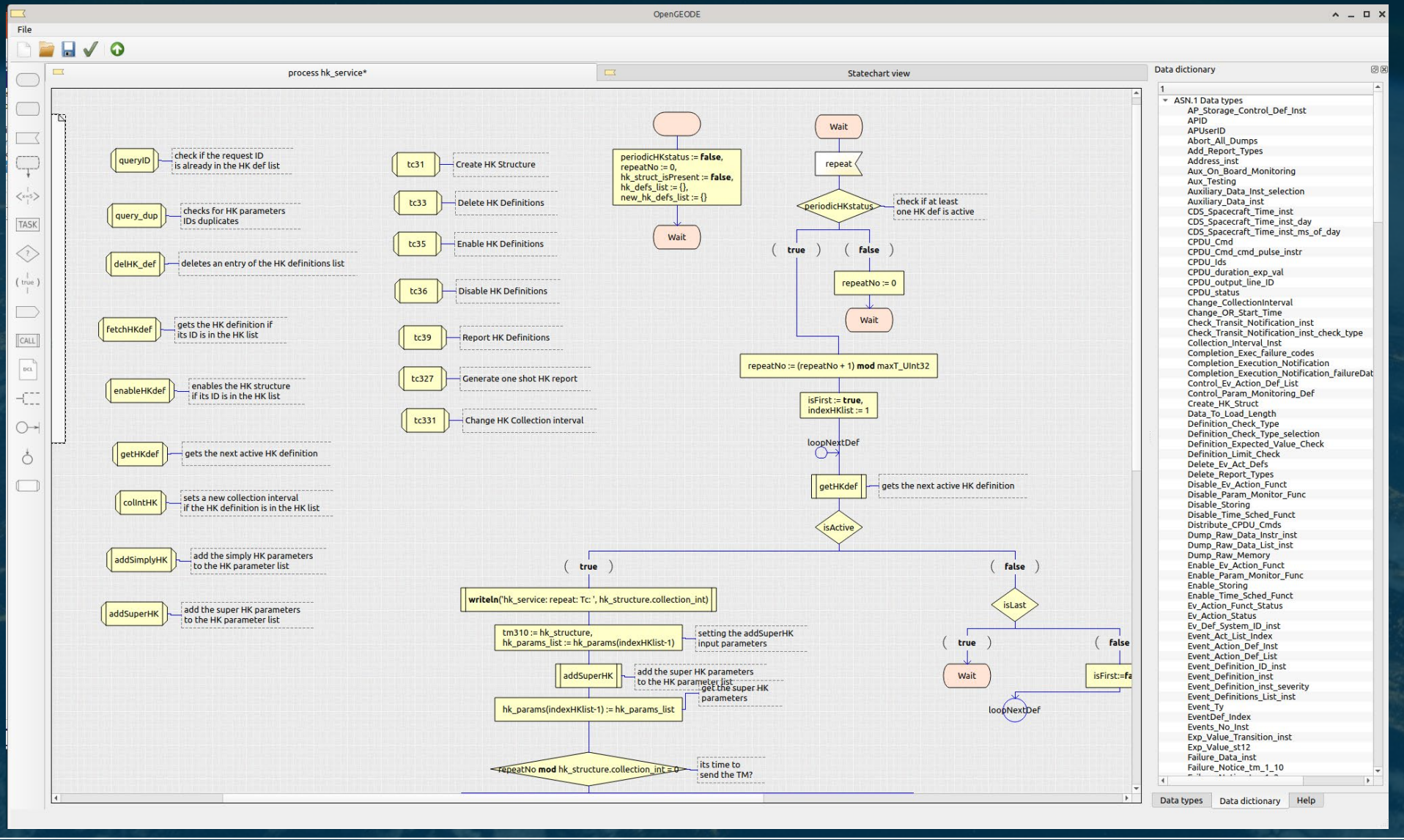
| No | Field | Comment | Present | Type | Constraint | Min Bits | Max Bits |
|---|---|---|---|---|---|---|---|



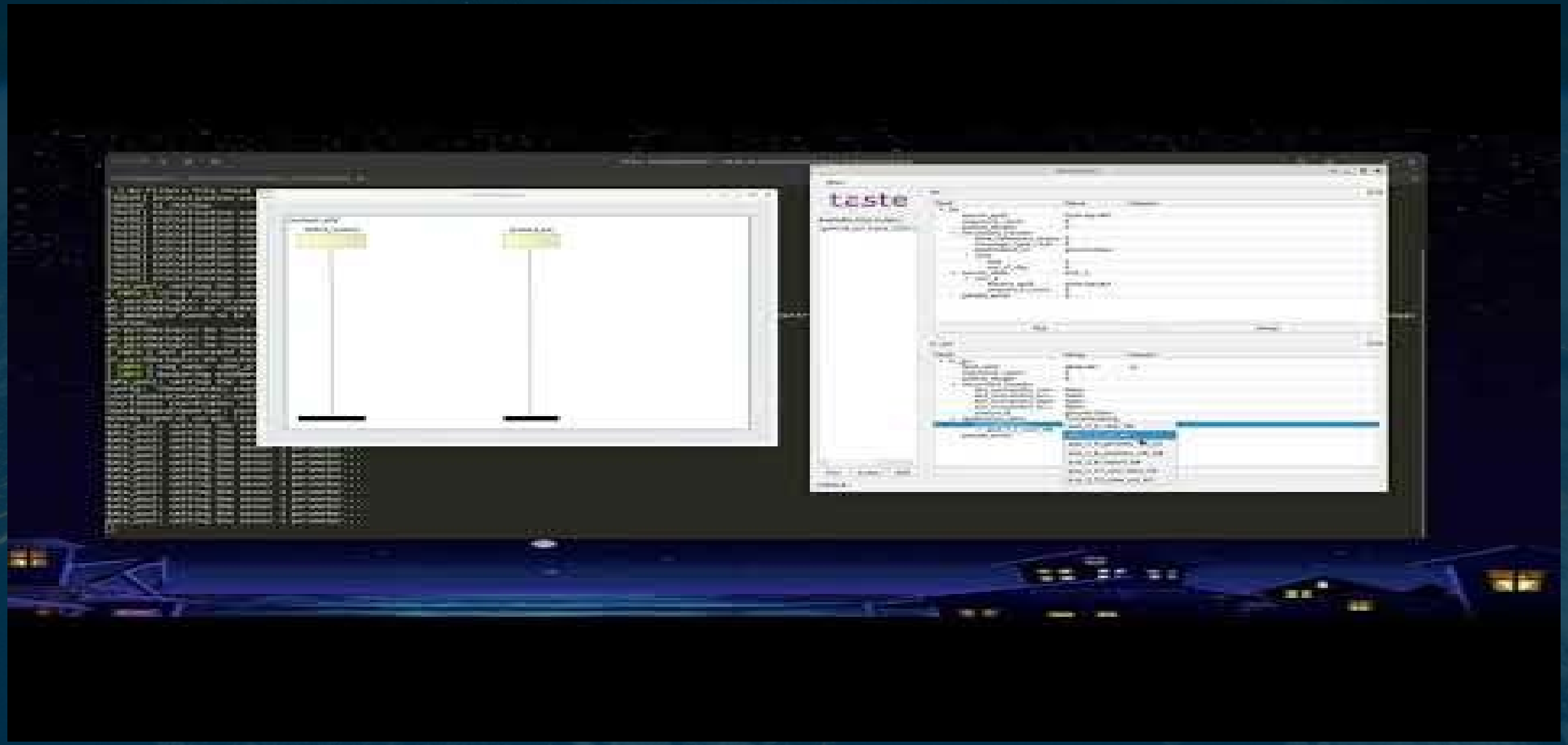
# opus to TASTE (2) - components



# Example: the Housekeeping service



# Demo of the housekeeping service



# Status and conclusion

- The PUS Execution Platform is work in progress
- The proof of concept includes only a few PUS services:
  - Housekeeping,
  - Monitoring,
  - Event-Actions

- **opus** will be integrated in taste by 2024 and allow:
  - Easy creation of new PUS services
  - Guided tailoring for PUS data
  - Automatic generation of TM/TC encoders for flight code
  - Automatic generation of a mission-tailored PUS library (including private services)

# Questions?

1. Quick introduction to TASTE
2. Demo
3. The SAVOIR Onboard Software Reference Architecture  (OSRA) and TASTE foundations
4. Modelling and working with a PUS-C execution platform