DCAAS: DATA COMPRESSION AS A SERVICE

FINAL PRESENTATION, 2022.11.04

VISIONSPACE

esa

AGENDA

1. Study Overview
2. POCKET+ / CCSDS-124
3. CompressionCache
4. Results and Summary
5. Discussion

1. STUDY OVERVIEW

MOTIVATION FOR DATA COMPRESSION

Reduction of transmission channel bandwidth

Less or shorter passes over ground stations needed, for lower operational costs

Reduction of buffer and storage requirement

More data can be stored without the need to downlink it, enabling more autonomous operation

Anomaly detection

Housekeeping can be gathered at a higher sampling rate leading to enhanced onboard anomaly detection

OUR BACKGROUND

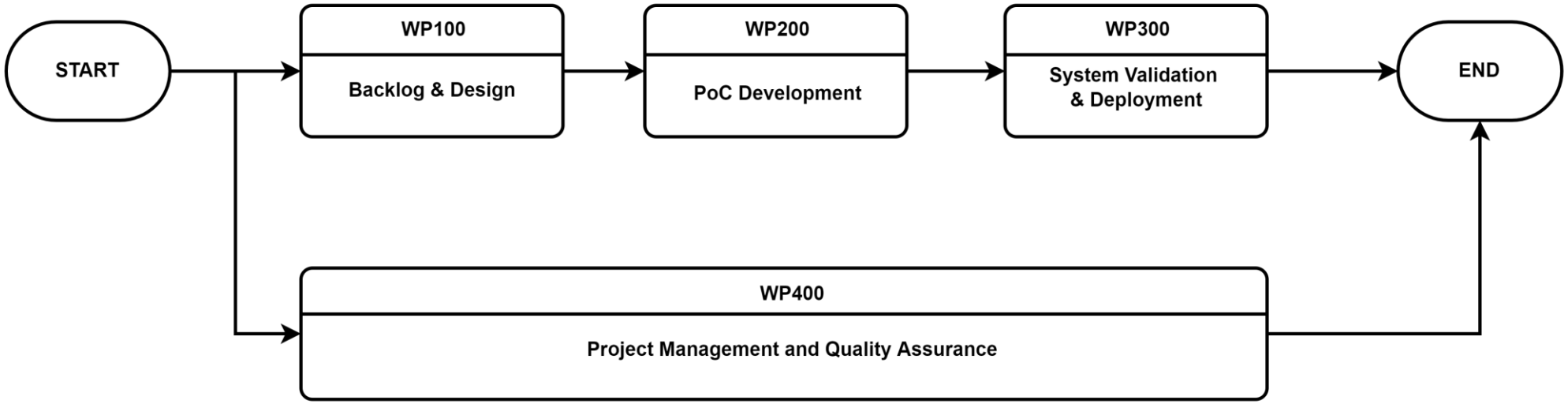
- VisionSpace is a NewSpace company located in Darmstadt.
- We implemented the SCOS2000 extension to support POCKET+ on OPS-SAT.
- Supported the OPS-SAT operations team in testing the extension.
- Contributed to the CCSDS standard review process for POCKET+.
- Implemented POCKET+ in VHDL for a rad-hard FPGA (NanoXplore NG-Medium).

GOALS

- Make telemetry data compression available to OPS-SAT experimenters.
- Demonstrate and document how to use POCKET+ it in an operational environment.
- Simplify the management of different contexts for POCKET+.
- Integrate the developed software components with existing space- and ground-based infrastructure.
- Disseminate findings, lessons learned, and share knowledge about the new CCSDS 124 standard.

SCHEDULE

11th of April 2022 → **4th of November 2022**

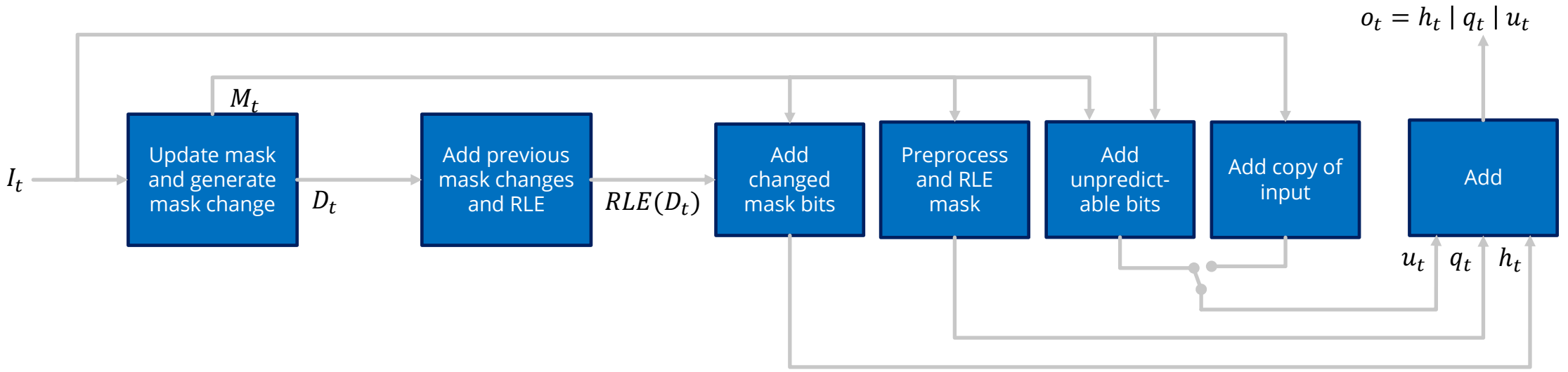


2. POCKET+ / CCSCS-124

CCSDS-124 HISTORY

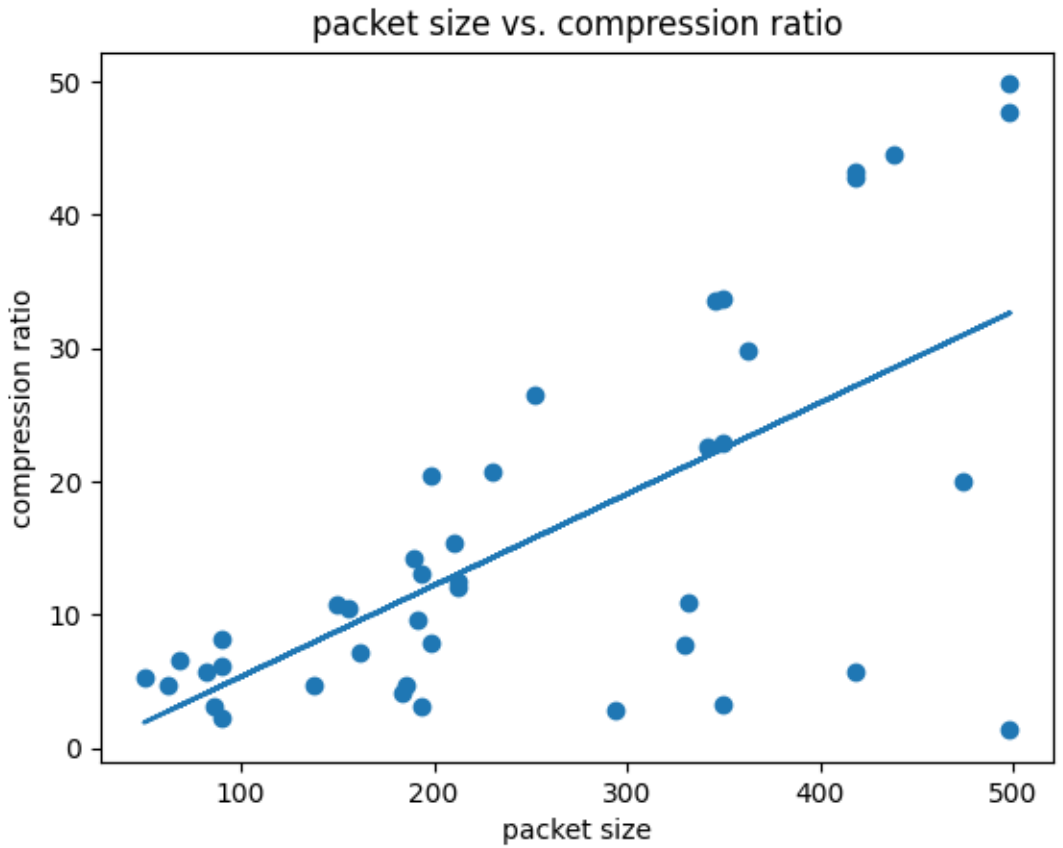
- 2011 – Patent filled for the POCKET algorithm
- 2015 – POCKET was successfully tested on ESA’s Proba-2 mission
- 2018 – POCKET+ with reduced complexity for ground support systems
- 2019 – POCKET+ was launched on ESA’s OPS-SAT mission
- 2019 – Patent filled for POCKET++ with enhanced compression performance
- 2020 – Draft CCSDS standard 124 “Robust Compression of Fixed-Length Housekeeping Data” is currently under review by the CCSDS data compression working group

COMPRESSION PROCESS



CCSDS-124 VALIDATION

- Latest version of CCSDS-124
- Compressor cross validation done
- Rosetta data set
- Robustness: 0
- New mask period: 10
- Decompressor not robust against packet loss

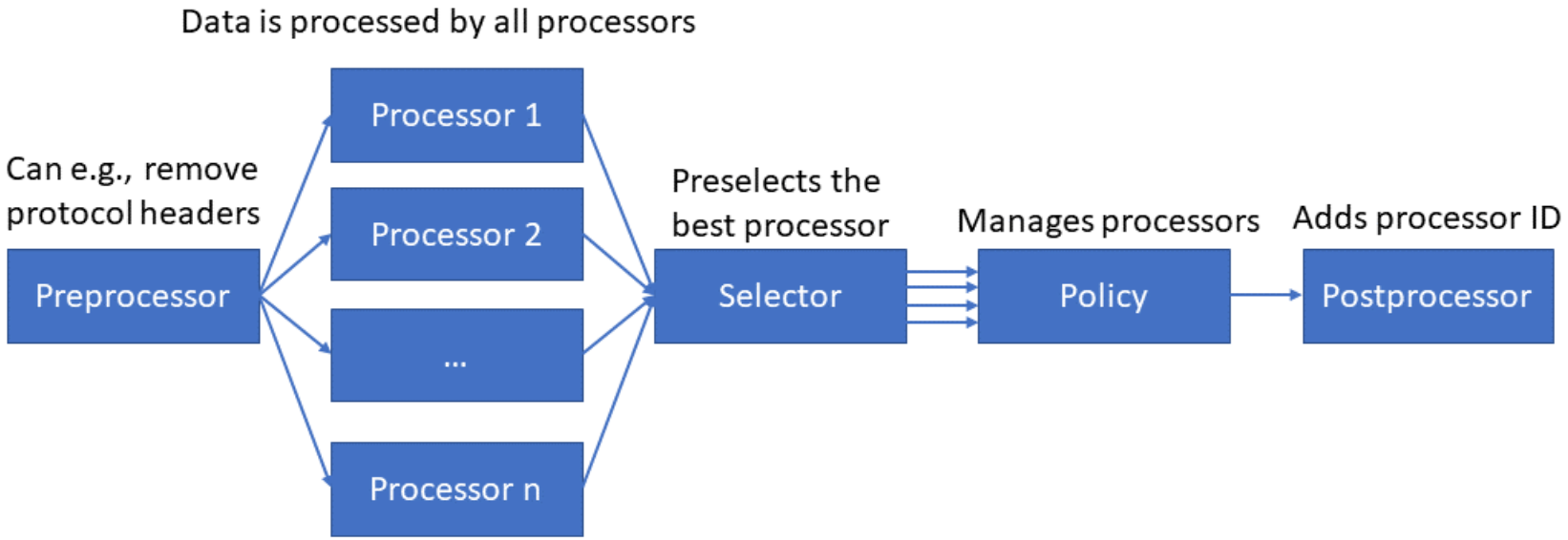


3. COMPRESSIONCACHE

EXPECTED OUTCOMES

- Space component able to compress data, managing multiple contexts simultaneously
- Ground component capable of decompressing and disseminating contextualized packets
- Develop a PoC prototype that:
  - Makes data compression available to the experimenters
  - Allows easy tuning of the compression performance
  - Gathering of compression performance metrics

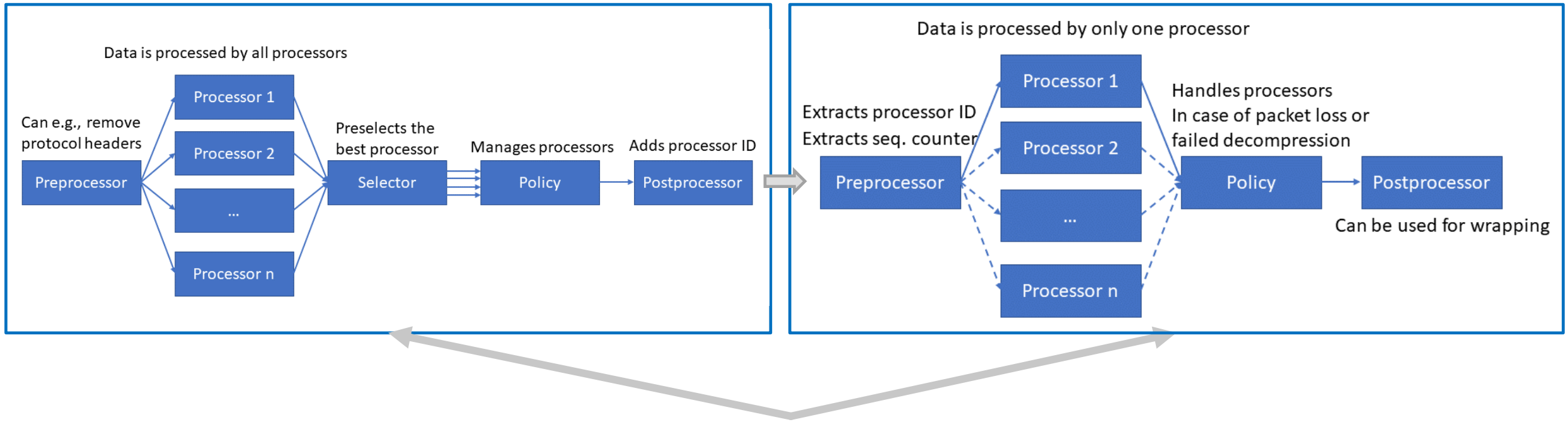
APPROACH – COMPRESSIONCACHE



COMPRESSIONCACHE DESIGN

Space-Component

Ground-Component



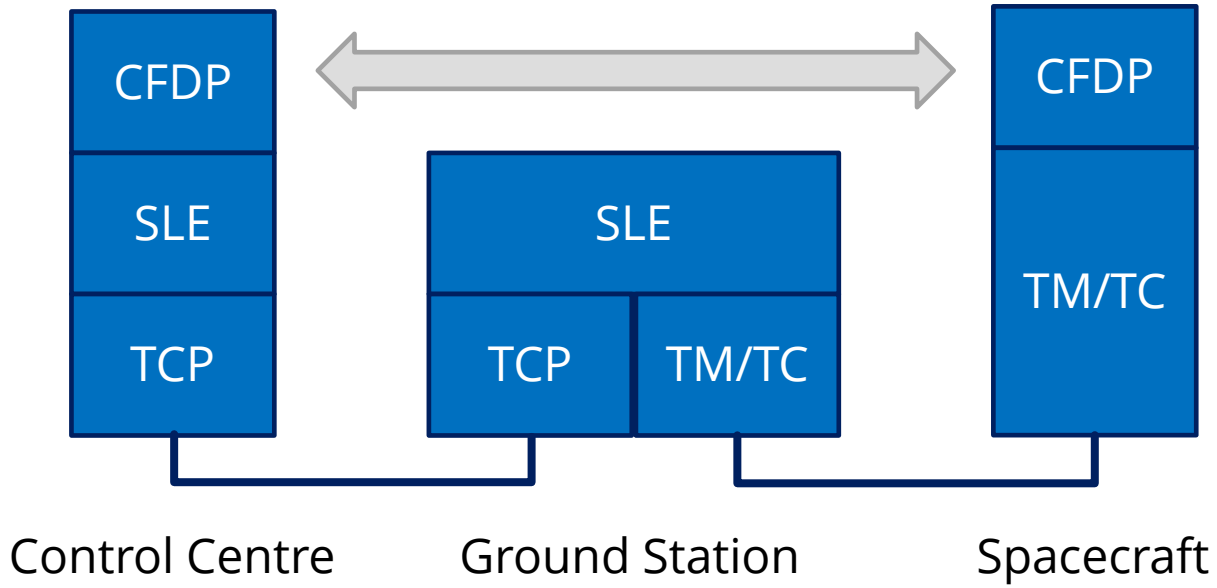
Configuration File

COMPRESSION CACHE FEATURES

- Separate compression and decompression applications
- Three runtimes: ZMQ-, File-, and TCP
- Three compression algorithms: Zstd, Snappy, and CCSDS-124 (POCKET+)

USE-CASE – FILE RUNTIME

- File based spacecraft operations
- Upload of control data
- Download of science data
- Download of housekeeping data

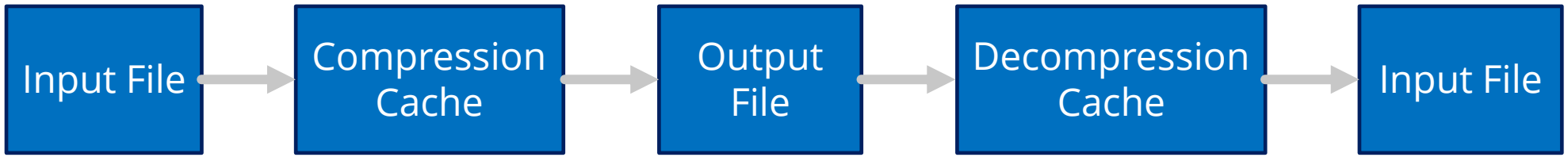


FILE-RUNTIME

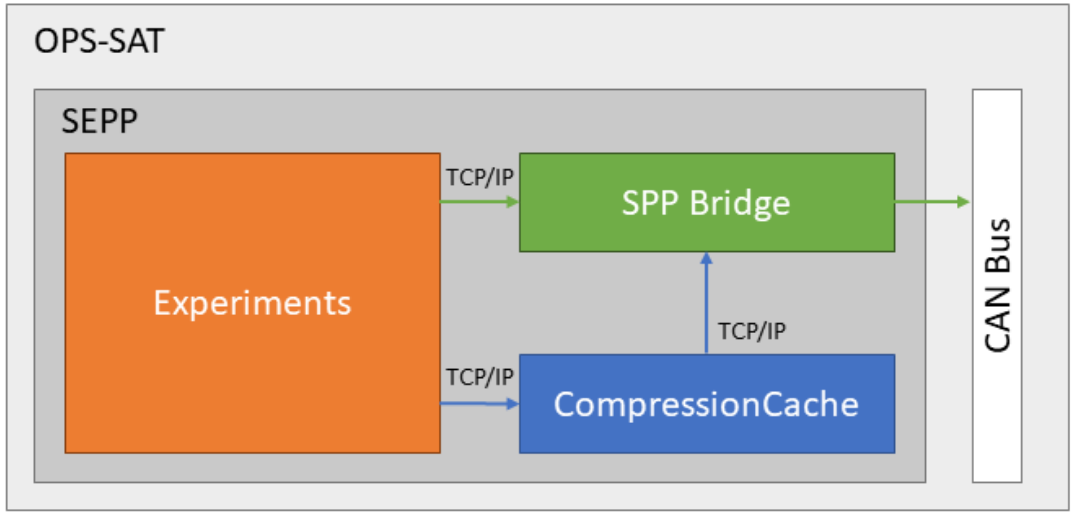
- Two separate applications
  - Ground and space element
- CCSDS Space Packets to compressed packets in a file
- Single Input Single Output (SISO)
- First In First Out (FIFO)
- No packet loss assumed

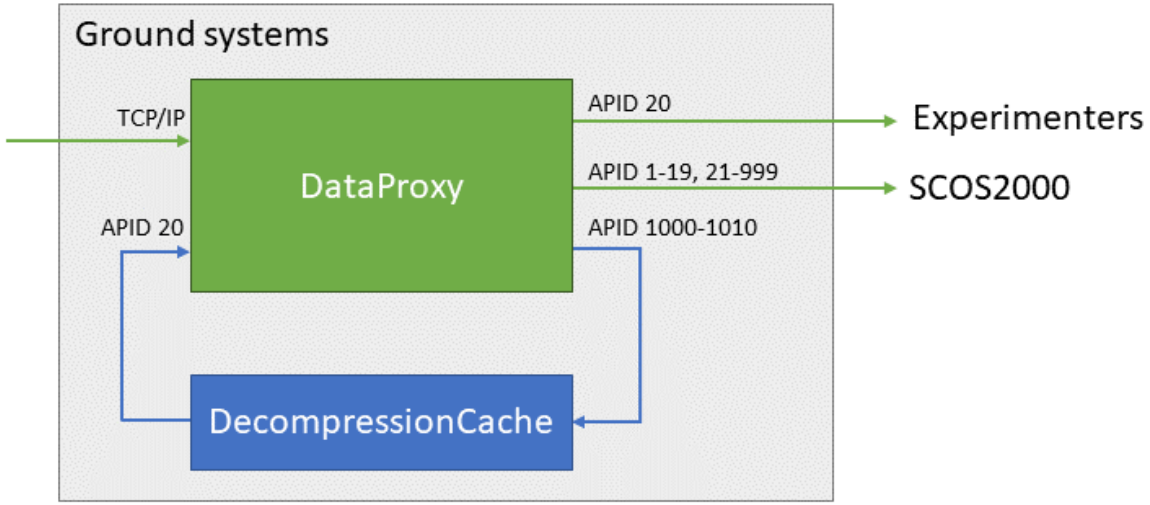
FILE-RUNTIME DEMO

- The file runtime runs the selected compression/decompression algorithm and writes the result to a new file. It can be selected as a command line argument with "runtime=file-input/file/name-output/file/name".
- To run the file runtime sequentially, using one terminal:
  - `./CompressionCache PPP=1 windowSize=10000 runtime=file-4638Packets.ccsds-output.compressed`
  - `./DecompressionCache PPP=1 windowSize=10000 runtime=file-output.compressed-output.decompressed`
  - `diff <(xxd 4638Packets.ccsds) <(xxd output.decompressed)`
  - `diff -y <(xxd 4638Packets.ccsds) <(xxd output.decompressed)`



USE-CASE – TCP RUNTIME



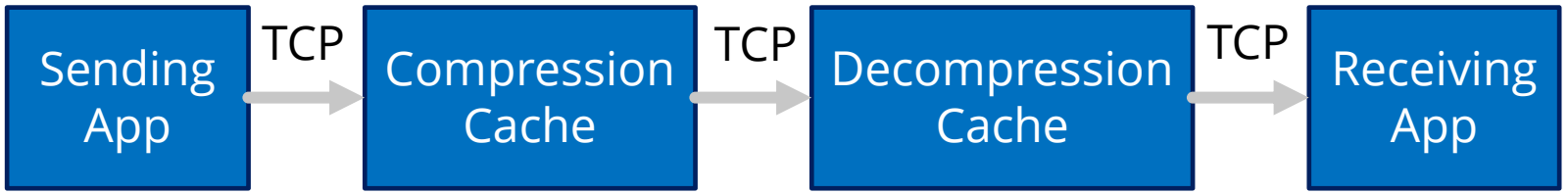


TCP-RUNTIME

- Two separate applications
  - Ground and space element
- Transfer compressed CCSDS Space Packets over TCP socket
- Single Input Single Output (SISO)
- First In First Out (FIFO)
- No packet loss assumed

TCP-RUNTIME DEMO

- The TCP runtime runs the selected compression/decompression algorithm over TCP sockets. It can be selected as a command line argument with "runtime=TCP".
- To run the TCP runtime, using 4 terminals:
  - `python3 python/tcp_pull_bin_spp_file.py`
  - `build/DecompressionCache PPP=1 windowSize=10000 runtime=TCP`
  - `build/CompressionCache PPP=1 windowSize=10000 runtime=TCP`
  - `python3 python/tcp_push_bin_spp_file.py`



UNIT TESTING

- Unit testing
  - Based on Gtest
  - 85 unit-tests implemented for PocketPlus
  - 116 unit-tests implemented for the CompressionCache
- Test coverage reports
- POCKET+ Cross validation
  - Fully validated based on Yellow Book for compression
  - Partially validated for decompression – without packet loss
  - Both data sets can be fully processed without a crash

TEST COVERAGE

LCOV - code coverage report

		Hit	Total	Coverage
Current view: top level				
Test: coverage.info.cleaned	**Lines:**	827	924	89.5 %
Date: 2022-10-05 11:57:36	**Functions:**	38	38	100.0 %

Directory	Line Coverage		Functions	
compressor	98.2 %	319 / 325	100.0 %	22 / 22
decompressor	83.2 %	435 / 523	100.0 %	7 / 7
utils	96.1 %	73 / 76	100.0 %	9 / 9

LCOV - code coverage report

		Hit	Total	Coverage
Current view: top level				
Test: coverage.info.cleaned	**Lines:**	498	524	95.0 %
Date: 2022-10-05 11:37:42	**Functions:**	131	155	84.5 %

Directory	Line Coverage		Functions	
include/compression	100.0 %	6 / 6	50.0 %	6 / 12
include/decompression	100.0 %	4 / 4	50.0 %	4 / 8
src/compression/cache	100.0 %	56 / 56	100.0 %	11 / 11
src/compression/policy	100.0 %	39 / 39	100.0 %	9 / 9
src/compression/postprocessor	95.0 %	38 / 40	100.0 %	9 / 9
src/compression/preprocessor	100.0 %	4 / 4	66.7 %	2 / 3
src/compression/processor	97.7 %	212 / 217	89.5 %	51 / 57
src/compression/selector	100.0 %	16 / 16	71.4 %	5 / 7
src/decompression/cache	90.2 %	37 / 41	100.0 %	8 / 8
src/decompression /postprocessor	100.0 %	4 / 4	66.7 %	2 / 3
src/decompression /preprocessor	100.0 %	11 / 11	66.7 %	2 / 3
src/decompression/processor	82.6 %	71 / 86	88.0 %	22 / 25

INTEGRATION TESTING

- Test procedures
- File-Runtime
  - Read from a file and compress to another file
  - Read compressed file, and write to decompressed output file
- TCP-Runtime
  - Listen on TCP port, compress and send to TCP port
  - Read compressed packets, decompress, and send to TCP port

4. RESULTS AND SUMMARY

SUMMARY

- Open-source CCSDS 124 implementation: https://github.com/visionspacetec/PocketPlus
- Compressor validated against CCSDS yellow book dataset
- Proven performance of latest version with real spacecraft data
- Demonstrated use of the new standard
- Covered multiple use-cases for spacecraft data compression

OUTLOOK

- Tests use-cases with OPS-SAT
- Improve POCKET+ performance
- Explore use-cases outside spacecraft operations
  - Drones
  - Industrial IoT

5. DISCUSSION

THANK YOU!

Milenko Starcik
milenko.starcik@visionspace.com

VisionSpace Technologies GmbH
Robert-Bosch-Strasse 7
64293 Darmstadt
Germany

twitter.com/visionspacetech

VISION SPACE