# ION Software-Defined Metadata Standard Report

Sanjeev Gunawardena, Air Force Institute of Technology

Alexander Rügamer, Fraunhofer Institute for Integrated Circuits IIS

Thomas Pany, Muhammad Subhan Hameed, Markel Arizabaleta, Universität der Bundeswehr München

Javier Arribas, Centre Tecnològic de Telecomunicacions de Catalunya

September 19, 2019

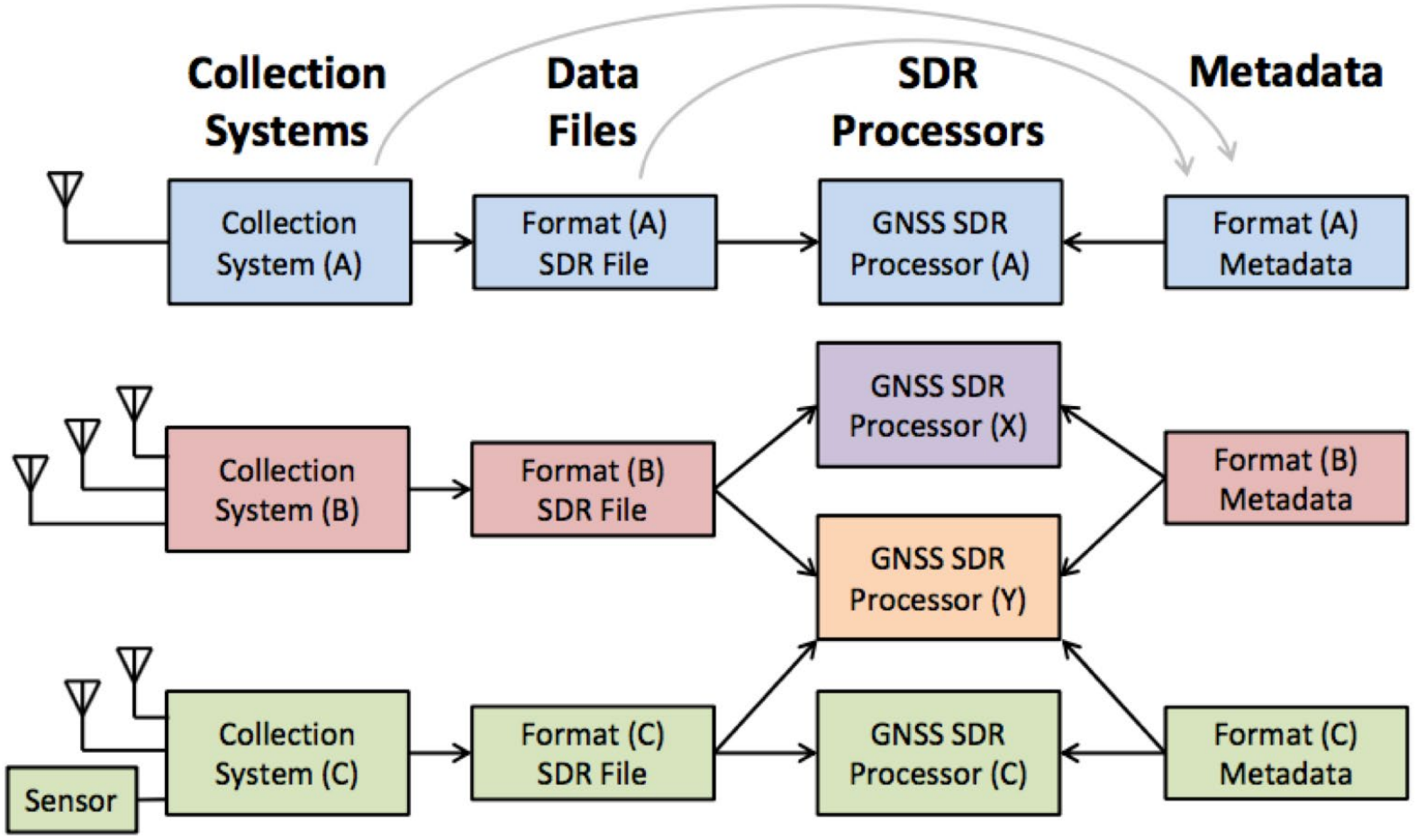
Miami, FL

# Background

- Proliferation of GNSS SDR technology in the past several years
  - Low-cost front-end hardware and data collection systems
  - Maturing GNSS SDR processors, receivers and software frameworks
- There was no established standard to convey GNSS SDR metadata
  - Existing metadata standards not well suited for needs of GNSS SDR and PNT community
- ION SDR Metadata Standard
  - Objective: To promote/support interoperability between GNSS SDR data systems and processors

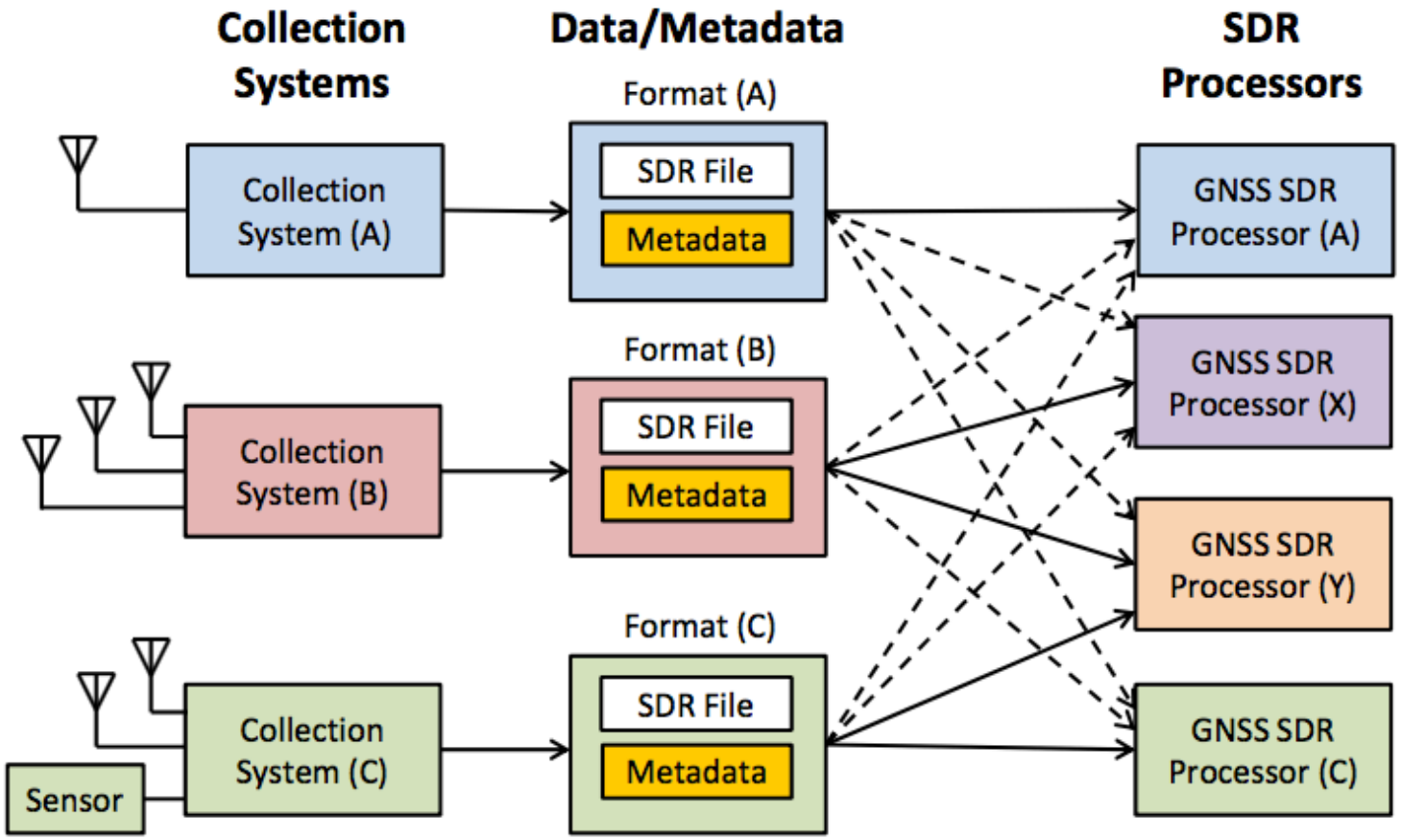
# The Problem

- Some front-end/DCS and SDR processors are bound to one another
- Ad hoc metadata exchange – prone to human error
- Does not promote interoperability
- Does not promote data/resource sharing and re-use

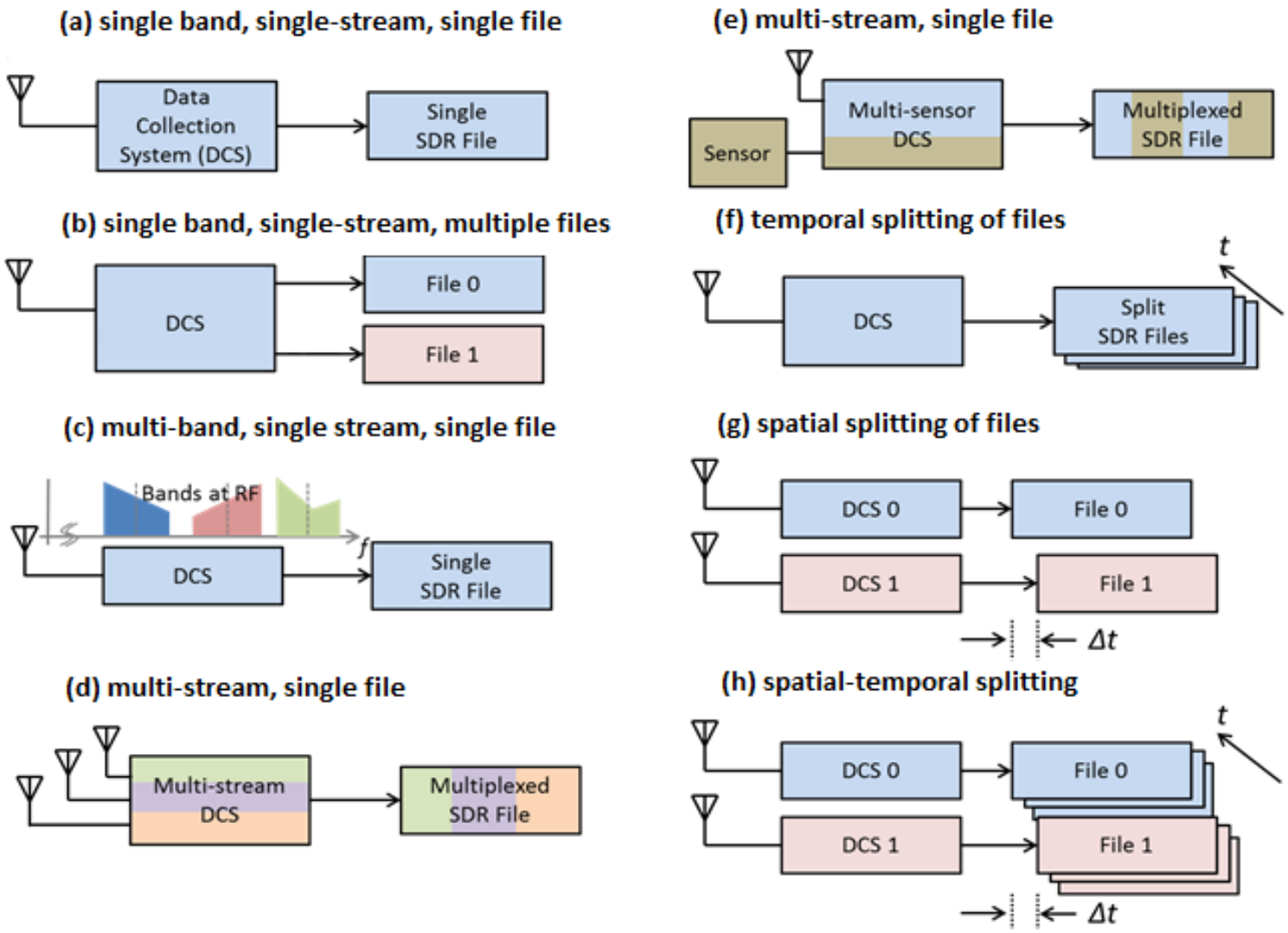


# Solution: ION Metadata Standard

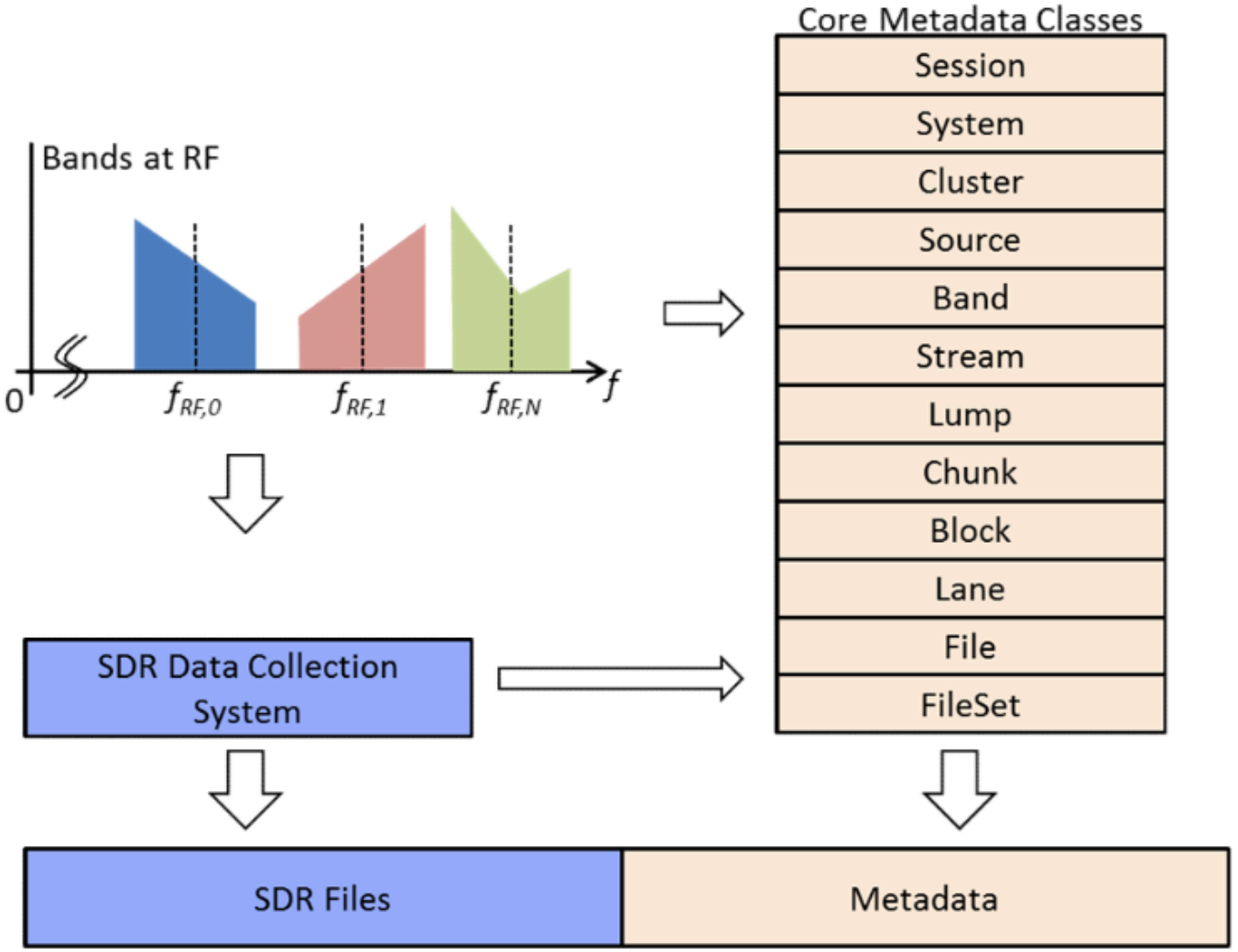
- Unambiguous transfer of all essential SDR metadata
- Standardization encourages vendors to support major formats
- Spurs community to develop open-source software handlers and plug-ins
- Promotes interoperability
- Promotes data portability, resource sharing and re-use



# System Topologies



# Metadata Domain Model



```xml
<?xml version="1.0" encoding="UTF-8"?>
<metadata xmlns="http://www.ion.org/standards/sdrwg/schema/metadata.xsd">
    <lane id="lane0">
        <block>
            <cycles>16368</cycles>
            <chunk>
                <sizeword>1</sizeword>
                <countwords>1</countwords>
                <endian>Undefined</endian>
                <padding>None</padding>
                <wordshift>Right</wordshift>
                <lump id="lump1">
                    <stream id="streamL1">
                        <ratefactor>1</ratefactor>
                        <quantization>2</quantization>
                        <packedbits>8</packedbits>
                        <alignment>Undefined</alignment>
                        <shift>Undefined</shift>
                        <format>IF</format>
                        <encoding>SMA</encoding>
                        <band id="bandL1">
                            <centerfreq format="MHz">1.5754200000000001e+03</centerfreq>
                            <translatedfreq format="Hz">4092000.000000e+00</translatedfreq>
                            <delaybias format="sec">0.0000000000000000e+00</delaybias>
                            <bandwidth format="MHz">4.0000000000000000e+01</bandwidth>
                        </band>
                    </stream>
                </lump>
            </chunk>
        </block>
    </lane>
    <system id="sys_GN3S">
        <comment format="text">GN3S module.</comment>
        <freqbase format="Hz">16368000.000000e+00</freqbase>
        <equipment>Equipment GN3S</equipment>
        <types>Undefined</types>
    </system>
    <session id="session0">
        <comment format="text">This is an example metadata specification for GN3S IF Data, one stream.</comment>
        <toa>1970-01-01T00:00:0.000000000Z</toa>
        <position lat="47.76471300" lon="-122.15612900" height="25.610"/>
        <contact>J. Doe</contact>
        <campaign>Example Campaign</campaign>
        <scenario>Example Scenario</scenario>
    </session>
    <file>
        <url>WB_fs16p368_4p092MHz_real_2bit_120s_001_052313_20-44.dat</url>
        <timestamp>2014-12-30T22:38:54.905999999Z</timestamp>
        <owner>Example Owner</owner>
        <lane id="lane0"/>
    </file>
</metadata>
```

# ← Example

## GN3S format

# Normative Reference Implementation

Consists of of two parts:

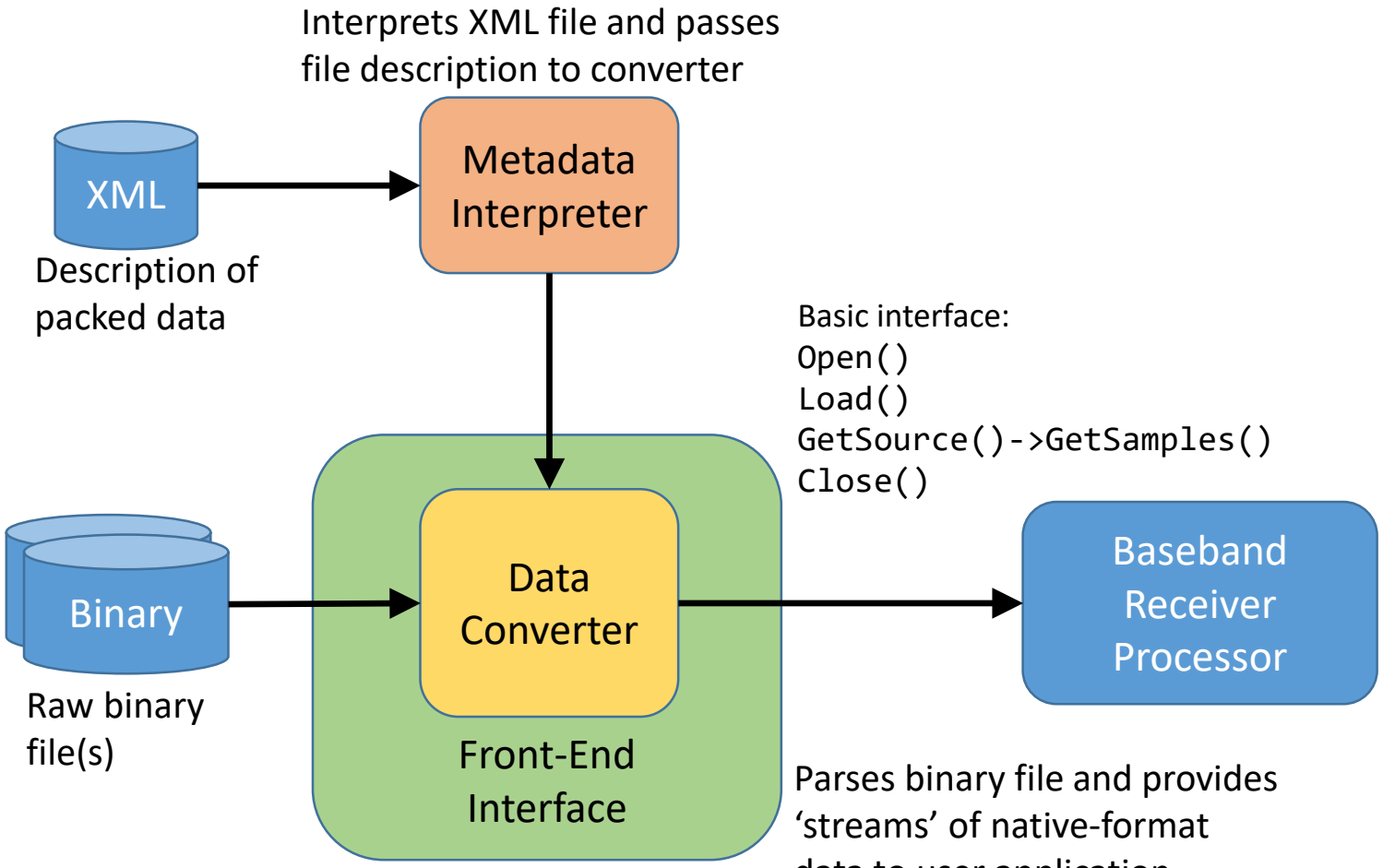
- A Metadata Interpreter:  libapi
  - Provides interface to read/write standard-compliant files
- A binary data converter:  libcnv
  - Provides an interface to read/convert GNSS IF data files

Open-source C++ project on GitHub:

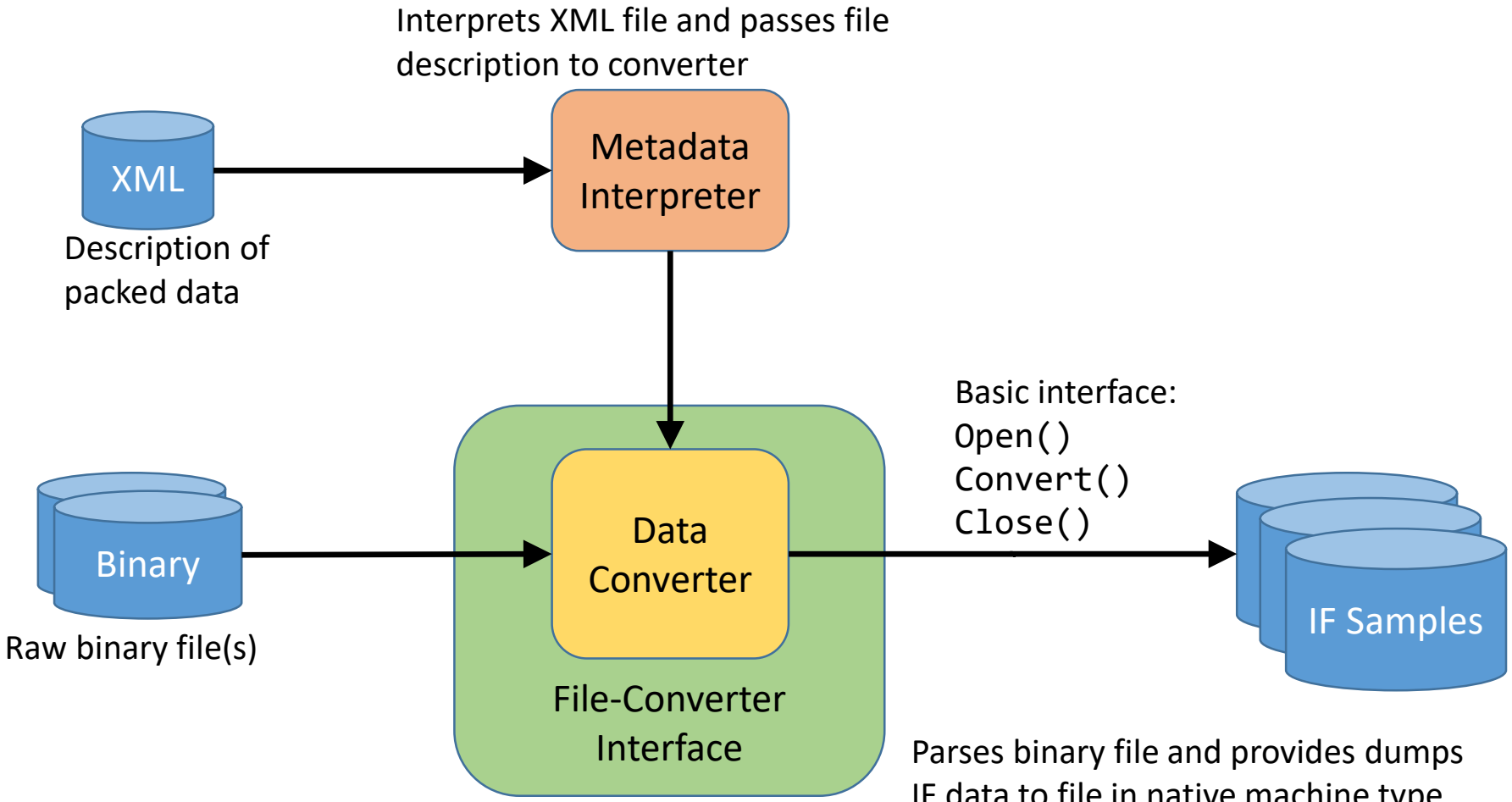
- https://github.com/IonMetadataWorkingGroup/GNSS-Metadata-Standard

- Cross platform: Windows, Linux, Mac OS

# Usage: in a software receiver



# Usage: as a file-converter



# Reference Data for Testing/Evaluation

- Fraunhofer Flexiband
- HackRF
- IFEN SX3
- Racelogic LabSat
- LimeSDR
- NTLab
- Nuand_BladeRF
- Ohio University TRIGR
- RTL SDR
- U. of Colorado GN3S
- SJTU SDR
- Ettus/NI USRP

http://sdr.ion.org/api-sample-data.html

# Adoption in Academia and Research

- Flexiband/GTEC front-end by Fraunhofer IIS
  http://www.iis.fraunhofer.de/flexiband
- Multi Sensor Navigation Analysis Tool (MuSNAT), by Universität der Bundeswehr München
  https://www.unibw.de/lrt9/lrt-9.2/software-packages/musnat
- GNSS-SDR
  https://gnss-sdr.org/
- pyChips: an open source python SatNav SDR for teaching and research

# Industry Adoption and Commercial Products

- Teleorbit GTEC© GNSS Radio Frequency Front-End
- Teleorbit MGSE (Record and replay)

  https://teleorbit.eu/en/satnav/gtec/

- IFEN SX3 GNSS Software Receiver

  https://www.ifen.com/products/sx3-gnss-software-receiver/

- TeleConsult Austria GIPSIE (GNSS simulator)

  https://www.tca.at/produkte/gnss-processing/gipsie

# Activities since September 2018

- September 24, 2018 – December 31, 2018
  - Completed second request for comments (RFC2)
  - Comments were mostly related to improvements to document

- September 2018 – July 2019
  - Minor revisions to normative software
  - RFC2 comments incorporated into Revision 0.4
  - Initial version of standard deemed complete
  - Additional data sets uploaded to sdr.ion.org:

# Activities since September 2018

- Aug 21 – Sept 4: Motion to approve the draft standard Rev. 0.4
  - Sent out to all voting members of the Working Group
  - Voting member: one individual from organization representing government, academia, industry

  - Results:

| | |
|---|---|
| Eligible to Vote | 57 |
| Approve | 26 |
| Disapprove | 0 |
| No Response | 31 |

(46% participation may have been due to email issues and summer vacations)

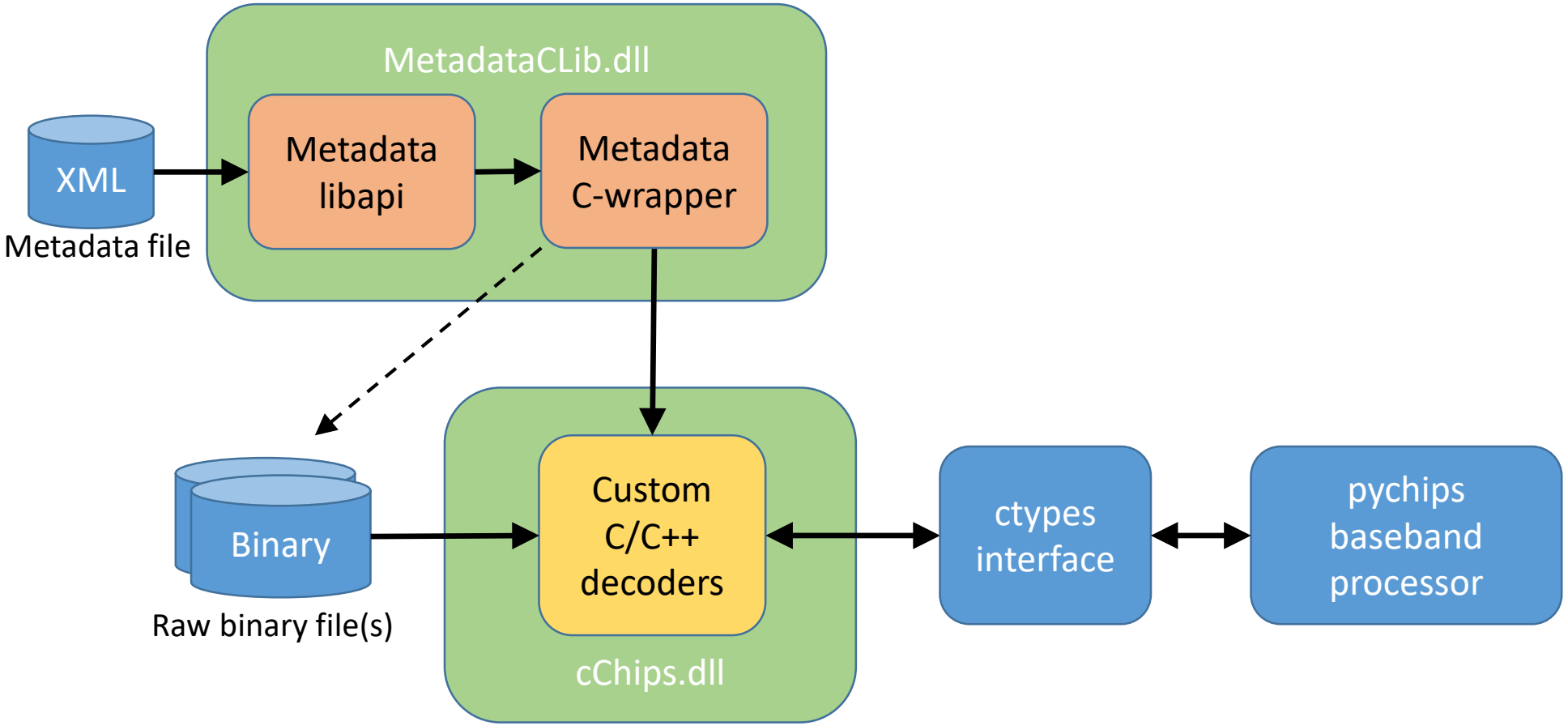
# Next Steps:

- Extend vote to include those who could not participate
- ION legal review
- Present to ION Council for adoption as formal ION Standard Rev. 1.0
- Additionally:
  - Promote standard adoption, initially in research/academic SDRs
  - Promote industry adoption, initially to output standard-compliant metadata files

# Standard Adoption Example 1: pyChips

- Python-based satnav SDR developed for teaching and research
- Open source: Apache 2.0 license
- Coming soon: GitHub repository access
- Last year: L1 C/A code processing using GN3S files (fs: 16 MHz, Bw: 4 MHz)
- This year: Multi constellation and multi-band:
  - GPS C/A, L5, L1C
  - Galileo E1, E5a, E5b
- File support: GN3S, LabSat 3 Wideband, TRIGR

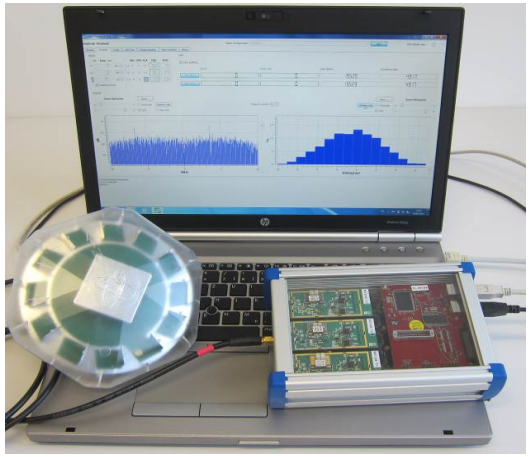
# ION Metadata interfacing to pyChips




Decodes samples to either int8, float64 (complex128) or both

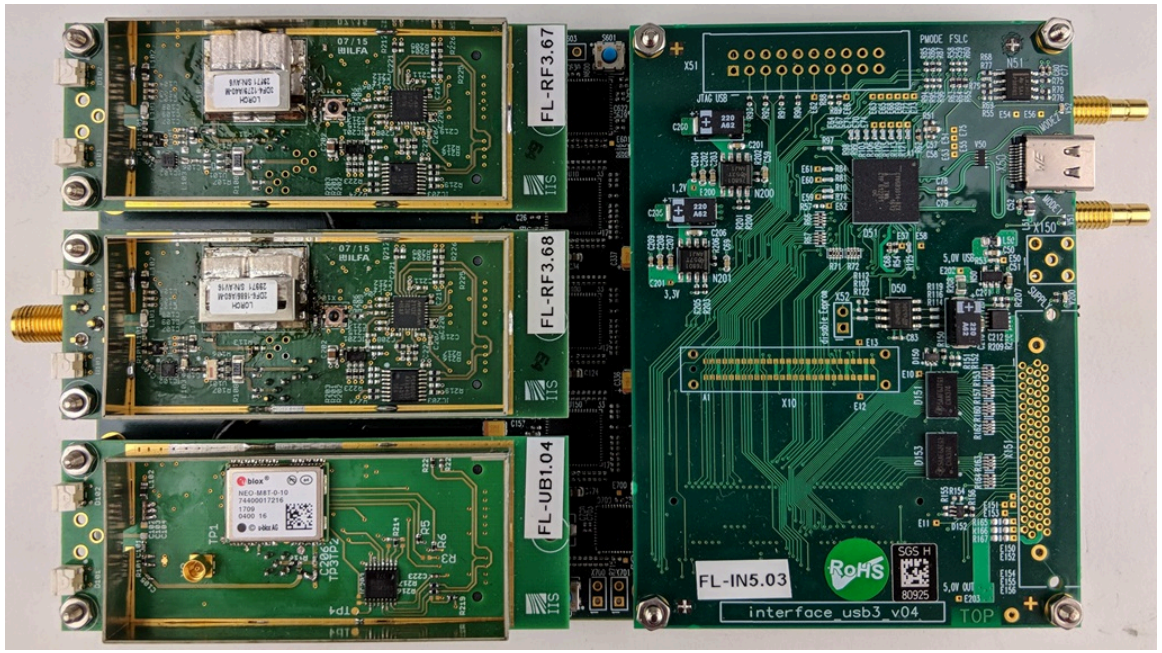
# Standard Adoption Example 2:
# Front-end samples with embedded non-sample data

# Front-end with embedded u-blox data



- Flexiband Front-end used for mobile recording campaigns
  - Raw data from up to three frequency bands via USB3 to multiplexed file on hard disk
  - Tagged with ION Metadata Standard
- Reference and assisted data required sometimes…
  - Solution: Internal “reference” by embedding a u-blox M8T receiver
- Benefits:
  - Absolute position and time reference w.r.t. the raw samples
  - Raw samples recording triggered by u-blox pps
  - U-blox ubx message provides GPS/Galileo/… information as assisted data

# Front-end with embedded u-blox data



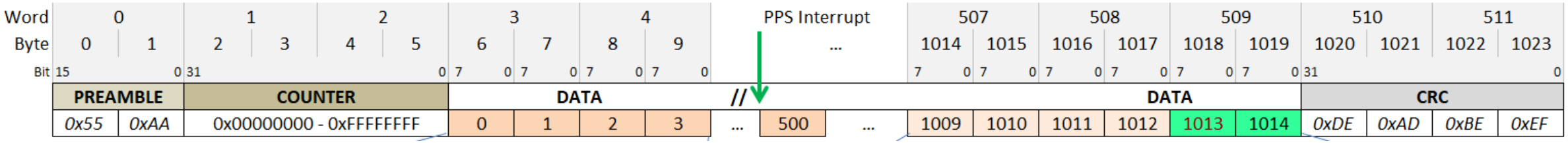

Flexiband USB front-end with:

- E1, 54 MHz BW, 4 bit I/Q, 81 Msps
- E6, 54 MHz BW, 4 bit I/Q, 81 Msps
- U-Blox NEO M8T

Objective:

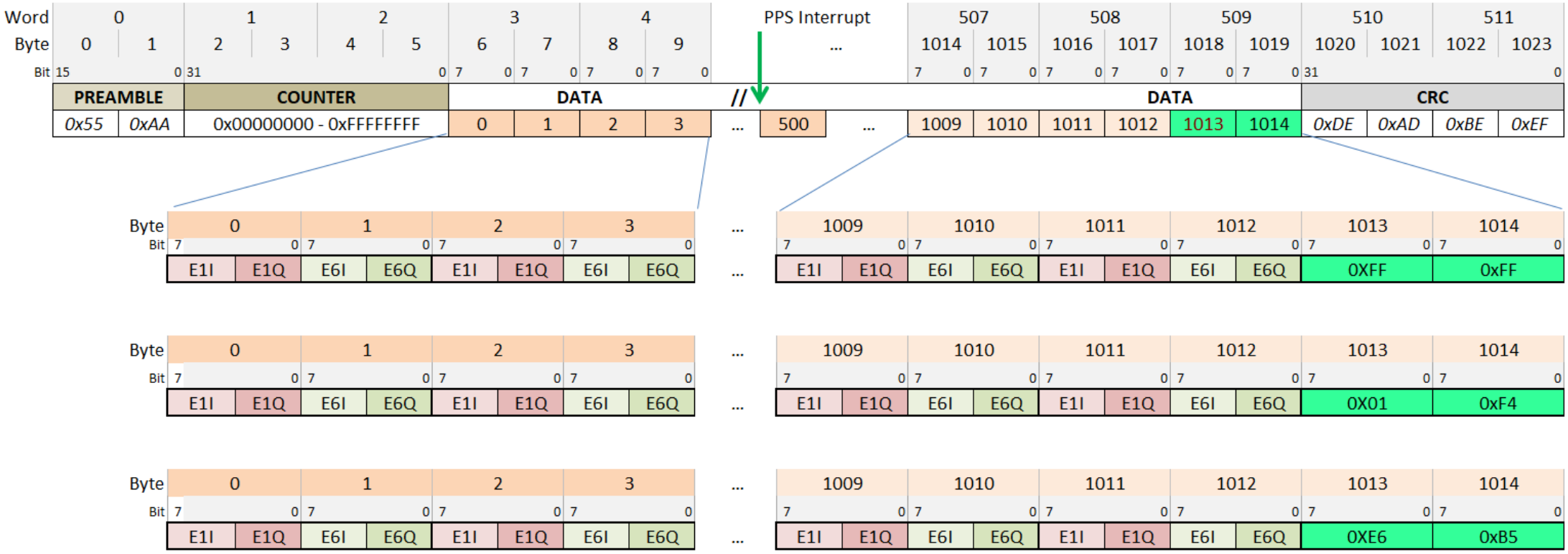
- Embedding of u-blox ubx output
- E.g. NAV-SVINFO, NAV-PVT and RXM-RAWX measurements, adding up to 580 byte if 10 satellites are received
- U-blox pps synchronized with raw data

# Front-end with embedded u-blox data

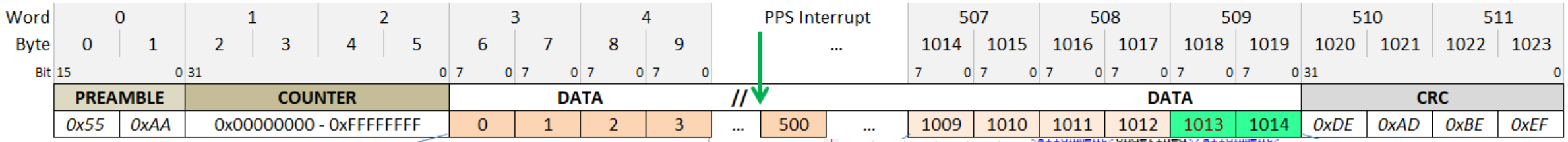



- Data streams are multiplexed to blocks of 1024 bytes length
  - 6 bytes header (2 bytes preamble, 4 bytes counter to detect packet loss)
  - 2x506 bytes for the E1 and E6 raw data
  - 6 bytes footer (2 byte data padding and 4 byte CRC)
- U-blox ubx and pps info embedded in footer (bytes 1018 and 1019)
  - Ubx message extracted in post processing
  - pps triggers measurements

# Front-end with embedded u-blox data



# Front-end with embedded u-blox data



| Word | 0 | | 1 | | 2 | | 3 | | 4 | | PPS Interrupt | 507 | | 508 | | 509 | | 510 | | 511 | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Byte | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | ... | 1014 | 1015 | 1016 | 1017 | 1018 | 1019 | 1020 | 1021 | 1022 | 1023 |
| Bit | 15 0 | | 31 0 | | | | 7 0 | 7 0 | 7 0 | 7 0 | | 7 0 | 7 0 | 7 0 | 7 0 | 7 0 | 7 0 | 31 0 | | | |
| | PREAMBLE | | COUNTER | | DATA | | | | | | // | DATA | | | | | | CRC | | | |
| | 0x55 | 0xAA | 0x00000000 - 0xFFFFFFFF | | 0 | 1 | 2 | 3 | ... | 500 | ... | 1009 | 1010 | 1011 | 1012 | 1013 | 1014 | 0xDE | 0xAD | 0xBE | 0xEF |

```
<?xml version="1.0" encoding="UTF-8"?>
<metadata xmlns="http://www.ion.org/standards/sdrwg/schema/metadata.xsd">
    <lane id="Data">
        <session id="0">
        <system id="Flexiband">
            <comment format="text">Front-end variant:II-4e PRS (E6abc/E1abc/uBlox)</comment>
            <freqbase format="MHz">8.1000000000000000e+01</freqbase>
            <equipment>Flexiband Multi-band receiver</equipment>
            <types>Processor</types>
        </system>
        <block>
            <cycles>506</cycles>
            <sizeheader>6</sizeheader>
            <sizefooter>6</sizefooter>
            <chunk>
                <sizeword>1</sizeword>
                <countwords>2</countwords>
                <endian>Undefined</endian>
                <padding>None</padding>
                <wordshift>Left</wordshift>
                <lump>
                    <stream id="E6abc_B3">
                        <comment format="text">Amp:162</comment>
                        <comment format="text">Antenna power:false</comment>
                        <comment format="text">AGC:false</comment>
                        <ratefactor>1</ratefactor>
                        <quantization>4</quantization>
                        <packedbits>4</packedbits>
                        <alignment>Undefined</alignment>
                        <shift>Undefined</shift>
                        <format>IQ</format>
                        <encoding>INT</encoding>
                        <band id="E6abc_B3">
                            <centerfreq format="MHz">1.2700000000000000e+03</centerfreq>
                            <translatedfreq format="kHz">8.7500000000000000e+03</translatedfreq>
                            <delaybias format="sec">0.0000000000000000e+00</delaybias>
                            <bandwidth format="Hz">0.0000000000000000e+00</bandwidth>
                        </band>
                    </stream>
                    <stream id="L1_E1abc_B1_G1">
                        <comment format="text">Amp:120</comment>
                        <comment format="text">Antenna power:false</comment>
                        <comment format="text">AGC:false</comment>
                        <ratefactor>1</ratefactor>
                        <quantization>4</quantization>
                        <packedbits>4</packedbits>
                        <alignment>Undefined</alignment>
                        <shift>Undefined</shift>
                        <format>IQ</format>
                        <encoding>INT</encoding>
                        <band id="L1_E1abc_B1_G1">
                            <centerfreq format="MHz">1.5800000000000000e+03</centerfreq>
                            <translatedfreq format="kHz">-4.5800000000000000e+03</translatedfreq>
                            <delaybias format="sec">0.0000000000000000e+00</delaybias>
                            <bandwidth format="Hz">0.0000000000000000e+00</bandwidth>
                        </band>
                    </stream>
                </lump>
            </chunk>
        </block>
    </lane>
    <file>
</metadata>
```

# Standard Adoption Example 3: MuSNAT

## RINEX obs Files for sdr.ion.org Sample Data Files

# RINEX obs Files for sdr.ion.org Sample Data Files

- Sample data files available at http://sdr.ion.org/api-sample-data.html processed with MuSNAT to generate RINEX observation files:

| DataSet | Signals Tracked | Position Fix | Epoch at pos fix | Total epochs |
|---|---|---|---|---|
| BladeRF | L1,E1 | yes | 50 | 61 |
| FHG | L1,L2,L5,E5a | no | - | 14 |
| HackRF | L1,E1 | no | - | 58 |
| IFEN | L1,E1 | yes | 54 | 1116 |
| LabSat_1bit | L1,E1 | yes | 60 | 87 |
| LabSat_2bit | L1,E1 | yes | 60 | 87 |
| LabSat_3bit | L1,E1 | yes | 60 | 87 |
| LimeSDR | L1,E1 | yes | 28 | 59 |
| NTLab | L1,L2,E1 | yes | 59 | 137 |
| RTLSDR | L1,E1 | yes | 46 | 58 |
| SiGe | L1,E1 | yes | 77 | 118 |
| SJTU_L1E1 | L1,E1 | yes | 56 | 58 |
| USRP | L1,E1 | no | - | 6 |

Upload next 1-2 week s

# File Format Comparison

## *GNSS-SDR* Configuration File

```
;######### GLOBAL OPTIONS ##################

GNSS-SDR.internal_fs_sps=20000000
GNSS-SDR.assist_dual_frequency_acq=true
GNSS-SDR.use_acquisition_resampler=true
GNSS-SDR.custom_year=2008

;######### SUPL RRLP GPS assistance configuration #####

GNSS-SDR.SUPL_gps_enabled=false
GNSS-SDR.SUPL_read_gps_assistance_xml=true
GNSS-SDR.SUPL_gps_ephemeris_server=supl.google.com
GNSS-SDR.SUPL_gps_ephemeris_port=7275
GNSS-SDR.SUPL_gps_acquisition_server=supl.google.com
GNSS-SDR.SUPL_gps_acquisition_port=7275
GNSS-SDR.SUPL_MCC=244
GNSS-SDR.SUPL_MNC=5
GNSS-SDR.SUPL_LAC=0x59e2
GNSS-SDR.SUPL_CI=0x31b0
```

- Structure/Class based variable assignment
- Uses open-source *GNSS-SDR* configuration API at back-end
- Implements both front-end (signal source) and receiver related configuration in the same file.
- Provides interface between user and *GNSS-SDR* processing chain
- User launches config file through command line

```
;######### SIGNAL_SOURCE CONFIG ############
SignalSource.implementation=Flexiband_Signal_Source
SignalSource.flag_read_file=true
SignalSource.signal_file=/datalogger/signals/Fraunhofer/L125_III1b_210s.usb
SignalSource.item_type=gr_complex
SignalSource.firmware_file=flexiband_III-1b.bit
SignalSource.RF_channels=3
;#frontend channels gain. Not usable yet!
SignalSource.gain1=0
SignalSource.gain2=0
SignalSource.gain3=0
SignalSource.AGC=true
SignalSource.usb_packet_buffer=128
```

Front-end related configuration

```
;######### GALILEO ACQUISITION CONFIG ############
Acquisition_1B.implementation=Galileo_E1_PCPS_Ambiguous_Acquisition
Acquisition_1B.item_type=gr_complex
Acquisition_1B.threshold=2.8
Acquisition_1B.use_CFAR_algorithm=false
Acquisition_1B.blocking=true
Acquisition_1B.doppler_max=5000
Acquisition_1B.doppler_step=125
Acquisition_1B.dump=false
Acquisition_1B.dump_filename=./acq_dump.dat
```

Receiver related configuration

## ION GNSS SDR XML File

```
<?xml version="1.0" encoding="UTF-8"?>
<metadata xmlns="http://www.ion.org/standards/sdrwg/schema/metadata.xsd">
  <lane id="GPS SPS Data / Galileo OS Data">
      <session id="0">
          <comment format="text">This is a 210 second long signal.</comment>
          <toa>XXXX-XX-XXXX:XX:XX</toa>
          <contact>Fraunhofer IIS</contact>
          <campaign>GPS/Galileo Signal Generator Sample</campaign>
          <scenario>210 seconds of L1, L2 and L5 multiplexed data</scenario>
      </session>
      <system id="Flexiband-1"/>
      <block>
          <cycles>253</cycles>
          <sizeheader>6</sizeheader>
          <sizefooter>6</sizefooter>
          <chunk>
              <sizeword>1</sizeword>
              <countwords>4</countwords>
              <endian>Little</endian>
              <padding>None</padding>
              <wordshift>Left</wordshift>
```

```
<system id="Flexiband-1">
    <comment format="text">Flexiband with L125 III-1b configuration</comment>
    <freqbase format="Hz">20000000e+000</freqbase>
    <equipment>Flexiband GNSS Front-end</equipment>
    <types>Processor</types>
</system>
<file>
    <url>L125_III1b_210s.usb</url>
    <timestamp>2014-12-30T22:38:54.905999999Z</timestamp>
    <lane id="GPS SPS Data / Galileo OS Data"/>
    <owner>Fraunhofer IIS</owner>
    <copyright>http://www.iis.fraunhofer.de/flexiband"</copyright>
</file>
```

```
<section name="Receiver3">
  <option name="AntennaId">0</option>
  <option name="BumpJump">false</option>
  <option name="BumpJumpIntegrationTime">1.0</option>
  <option name="BumpJumpMinDistance">0.5</option>
  <option name="BumpJumpThreshold">1.05</option>
  <option name="CarrierPhaseAlways">false</option>
  <option name="CodeMultiPathStdDev">1.0</option>
  <option name="CoherentFraction">1</option>
  <option name="CoherentFractionSlave">1</option>
  <option name="CoherentIntegrator">0</option>
  <option name="CoherentIntegratorSlave">2</option>
  <option name="CorrelatorMode">0</option>
  <option name="CorrelatorModeSlave">0</option>
  <option name="CorrelatorType">1</option>
```

- XML-schema based file format.
- Uses open-source ION SDR C++ API at back-end
- Only implements front-end related (signal source configuration).
- Receiver related configuration is implemented within the GNSS SDR processor.
- Provides interface between a generic GNSS SDR processor and sampled front-end data stream.
- User launches GNSS SDR processor which invokes ION SDR XML file.

Front-end related configuration (implemented in ION SDR SML file)

Receiver related configuration (implemented in MuSNAT config file which is a separate file)

# MuSNAT and *GNSS-SDR* Configuration

| | | L1 | L2 | L5 | E1 | E5a |
|---|---|---|---|---|---|---|
| MuSNAT | **Bandwidth (narrow) (Hz)** | | | | | |
| | DLL | 1 | 1 | 1 | 1 | 1 |
| | PLL | 15 | 15 | 15 | 15 | 15 |
| | FLL | 1 | 1 | 1 | 1 | 1 |
| | **Bandwidth (wide) (Hz)** | | | | | |
| | DLL | 3 | 3 | 3 | 3 | 3 |
| | PLL | 20 | 20 | 20 | 20 | 20 |
| | FLL | 10 | 10 | 10 | 10 | 10 |
| | **Correlator Spacing (chips)** | 0.2 | 0.1 | 0.5 | - | 1 |

- **Configured to track :**
  - GPS L1 C/A , L2 CM, L5
  - Galileo E1 BC, E5a and E5b
- **Wave-form based tracking mode**
  - Early minus late difference computed at the level of the replica signals

| | | L1 | L2 | L5 | E1 | E5a |
|---|---|---|---|---|---|---|
| GNSS-SDR | **Bandwidth (narrow) (Hz)** | | | | | |
| | DLL | 0.5 | - | 0.5 | 0.5 | 0.5 |
| | PLL | 5 | - | 6 | 5 | 7.5 |
| | FLL | - | - | - | - | - |
| | **Bandwidth (wide) (Hz)** | | | | | |
| | DLL | 2 | 0.25 | 1.5 | 0.75 | 1.5 |
| | PLL | 35 | 2 | 20 | 15 | 20 |
| | FLL | 10 | 10 | 4 | 10 | 4 |
| | **Correlator Spacing (chips)** | - | 0.5 | 0.5 | 0.15 | 0.5 |

- **Configured to track :**
  - GPS L1 C/A , L2 C, L5
  - Galileo E1 BC, E5a and E5b

# *GNSS-SDR* – MuSNAT Zerobaseline Test

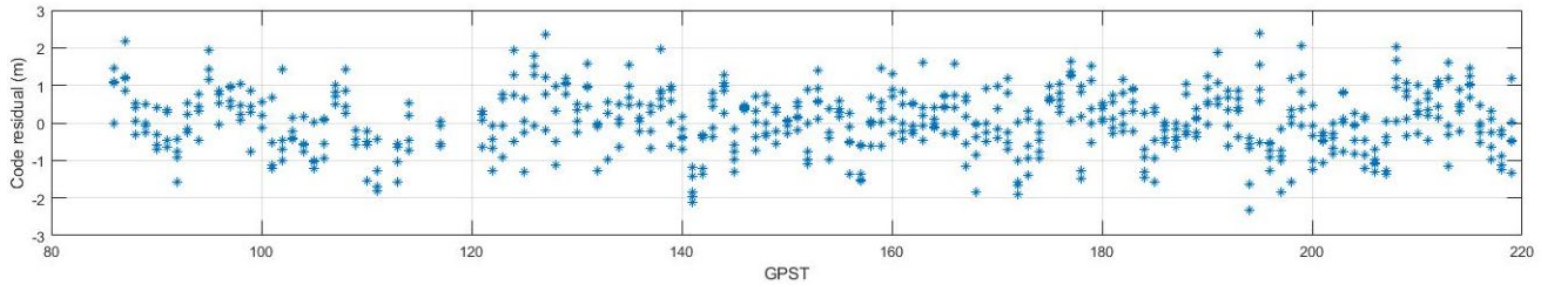
**Experimental Set-up**

- **Dataset**
  - 210s long Fraunhofer data set available at ftp://ftp.iis.fraunhofer.de/flexiband/reference-data/L125_III1b_210s.usb.zip
  - Recorded using Flexiband front-end with 3 frequency bands to track L1/E1, L2 and L5/E5a

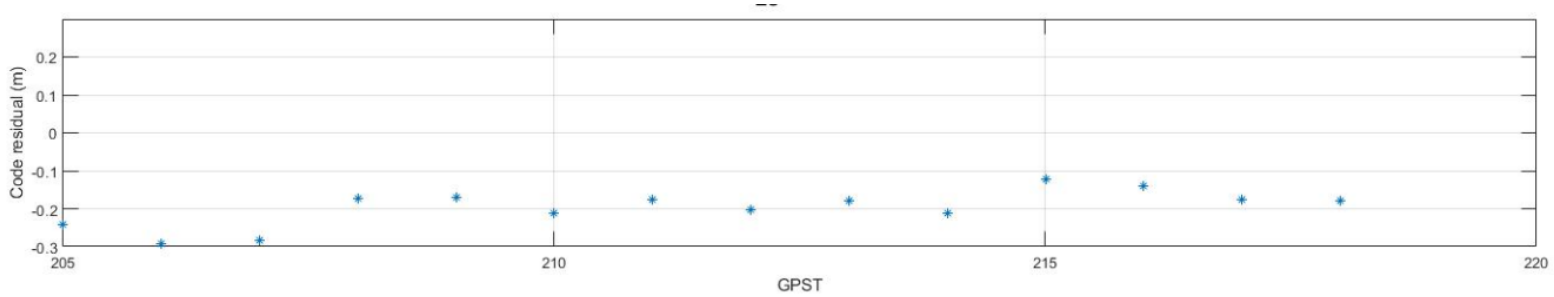
- **GNSS SDR – MuSNAT Zerobaseline Test**
  - Dataset processed separately with GNSS SDR and MuSNAT to generate rinex observation files
  - Post-processing done in RTKLib configured in Fixed mode with the GNSS SDR set as reference receiver
  - Reference coordinates taken from campaign recording snapshot available on Fraunhofer website
- **Results**
  - Position fix obtained for 83 % of time duration
  - Code and phase residuals obtained for E1 and E5a

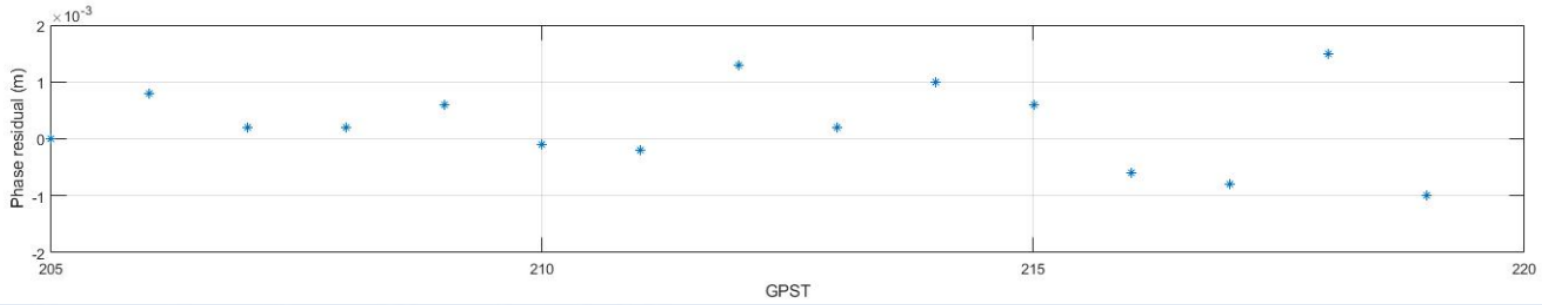
# Code and Phase Residuals (Preliminary results)

L1 C/A

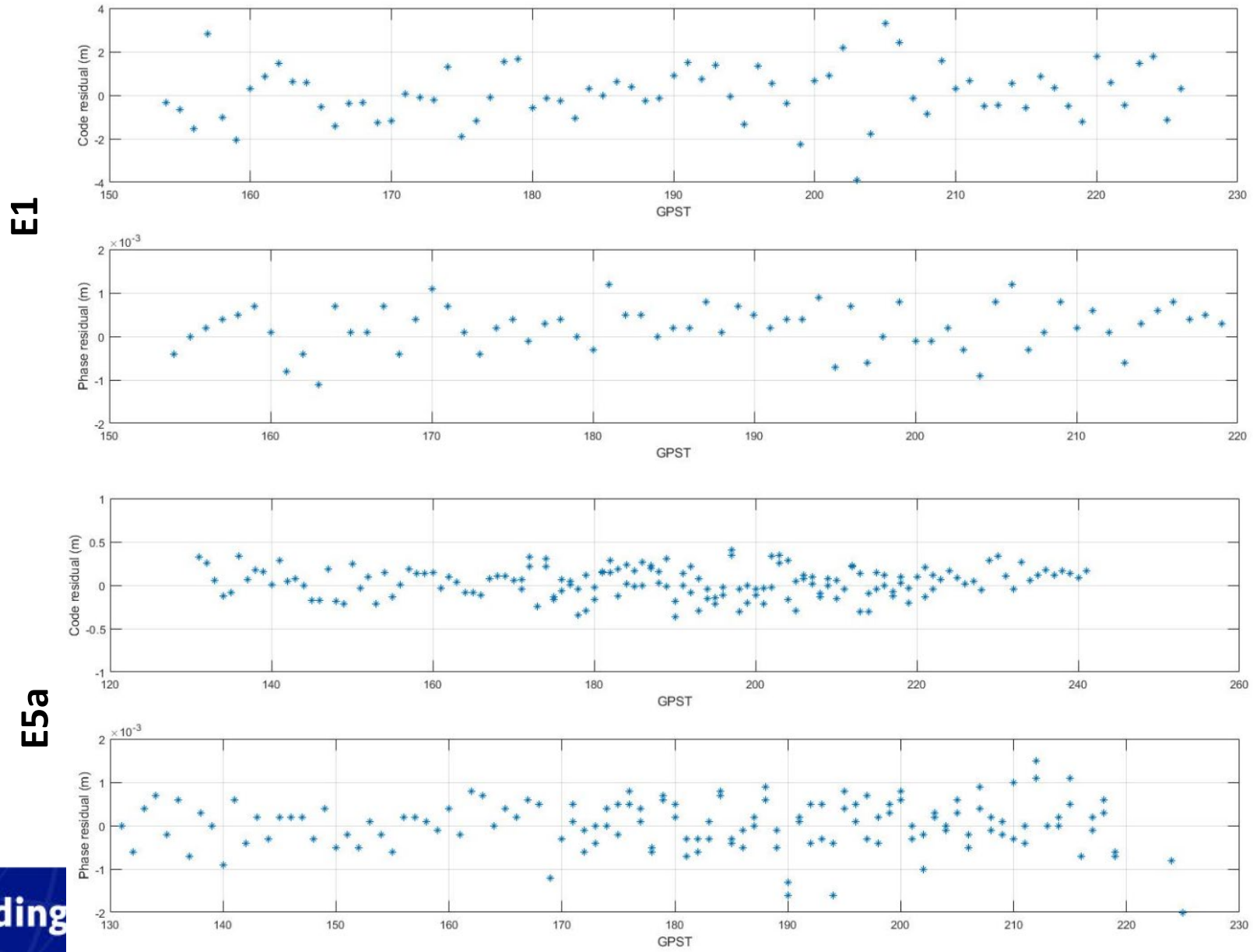



L5



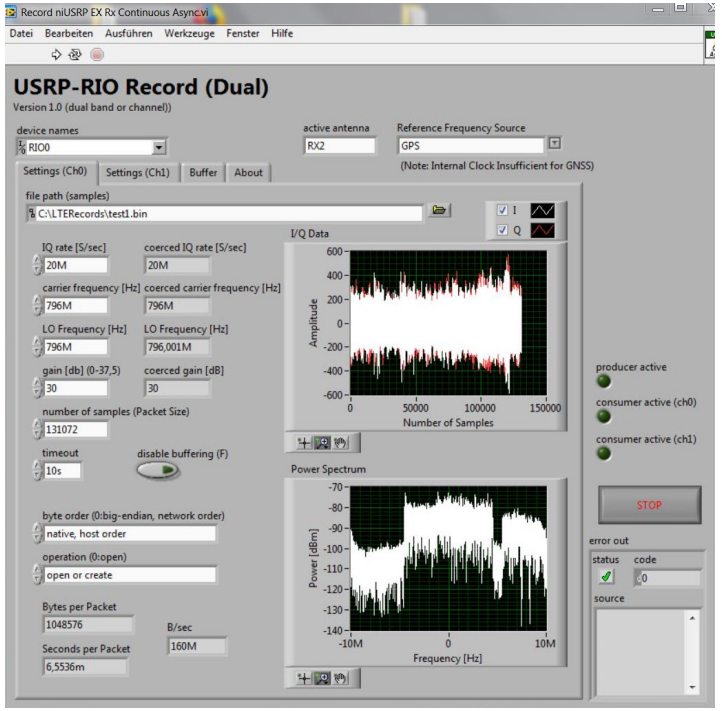



# Code and Phase Residuals (Preliminary results)



# LTE Data Set (Downlink Signal)

```xml
<?xml version="1.0" encoding="UTF-8"?>
<metadata xmlns="http://www.ion.org/standards/sdrwg/schema/metadata.xsd">
    <system id = "USRP">
        <freqbase format="MHz">10.0</freqbase>
    </system>
    <band id="800">
        <centerfreq format="Hz">796000000.00000</centerfreq>
        <translatedfreq format="Hz">0.0</translatedfreq>
    </band>
    <lane id="LTE800">
        <system id="USRP"/>
        <block>
            <cycles>0</cycles>
            <chunk>
                <sizeword>2</sizeword>
                <countwords>2</countwords>
                <endian>Little</endian>
                <lump>
                    <stream id="Chan0">
                        <ratefactor>1</ratefactor>
                        <quantization>16</quantization>
                        <packedbits>32</packedbits>
                        <format>IQ</format>
                        <encoding>TCA</encoding>
                        <band id="800"/>
                    </stream>
                </lump>
            </chunk>
        </block>
    </lane>
    <file>
        <url>1109195</url>
        <timestamp>2019-09-05T09:53:00Z</timestamp>
        <lane id="LTE800"/>
    </file>
```




MuSNAT User: l93athpa

Configuration | ControlLog | ReceiverLog | EpochDataHistory | Tracking Status | Acquisition Status | IF Sample Status

| NumStreams | AntennaId | FreqRf | FreqIf | Bias | StdDiv | RootMeanSquare | SampleRate |
|---|---|---|---|---|---|---|---|
| 2 | 0 | 796000000 | 5000000 | -13.83 | 16463.62 | 16463.63 | 20000000 |
| 2 | 0 | 1845000000 | 5000000 | 0.07 | 2037.24 | 2037.24 | 20000000 |
| | | | | | | | |

# Summary

- Initial work on the standard deemed complete!
- Few more steps to formal adoption by ION:
  - Majority vote to approve next step
  - ION legal review
  - Adoption by ION Council
- After formal adoption standard version will be set to v1.0
- We will continue to promote and help with adoption
- We will continue to revise/maintain software codebase and make minor revisions to document as needed
- Further promotion of sdr.ion.org for SDR related data exchange
- Thank you to everyone who contributed!

Please stay for Working Group meeting. Everyone is invited!

# Working Group Meeting

Minutes of Meeting

# Minutes of Meeting

- The meeting of the working group started immediately after session F4 on Thursday Sept. 19, 2019 at 17:30 in the same room Orchid. The attendee list is shown in the last slide of this presentation.
- Minutes and attendee list were taken by T. Pany .
- S. Gunawardena explained the voting results (first round) to approve the draft standard and outlines that for legal reasons a participation level of 50 % should be achieved. He emphasizes the support of the ION office as this could become the second standard that will be created under the ION umbrella. Standardization efforts will also be supported by the ION in future.
- The positive voting results (no objections) was accepted by the audience as well as reasons for low participation (email issues, suboptimal chosen time period for first voting). No objections against an extent of voting period were raised.
- Discussion on how many institution have been involved. S. Gunawardena outlines that initially ~80 have shown their interest. Later this number dropped to ~20.
- Question arose, if there is a MATLAB decoder for the standard available. -> yes, within the pyChips GNSS SDR of S. Gunawardena via a Mex wrapper
- Question arose, if a direct connection to GNU-Radio/USRP exists? No direction connection exists, but standard can applied within that framework
- Discussion on the possibility to directly assign coding task to the enlisted GitHub developers. No conclusion on that point.

# List of attendees

| Name | Organisation |
|---|---|
| Jan Wendel | Airbus |
| Mark Psiaki | Virginia Tech |
| Alexander Rügamer, Johannes Rossouw van der Merwe | Fraunhofer IIS |
| Roman Lesjak | Joanneum Research |
| Alex Tkatch | Rohde & Schwarz |
| Sandeep Chojar | Sierra Nevada Corporation |
| Lazaros Kapsias | Loctio |
| Dennis Akos | CU Boulder |
| Todd Kawakami | NGA |
| Johnathan York, Peter Hokanson | University of Texas |
| Zhen Zhu | East Carolina University |
| Fabio Dovis, Alex Minetto, Caner Savas | Politecnico di Torino |
| Aiden Morrison | SINTEF Norway |
| Michael Braasch | Ohio University |
| Mark Carroll | AFRL |
| Thomas Pany, Jürgen Dampf | UniBw M |
| Sanjeev Gunawardena | AFIT |