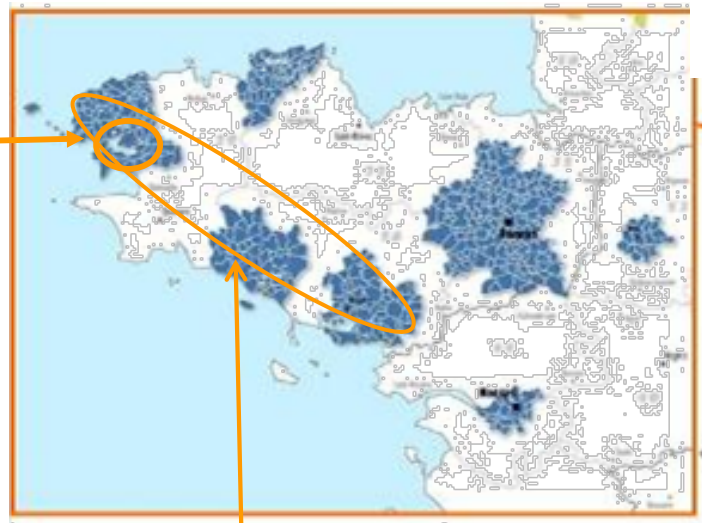
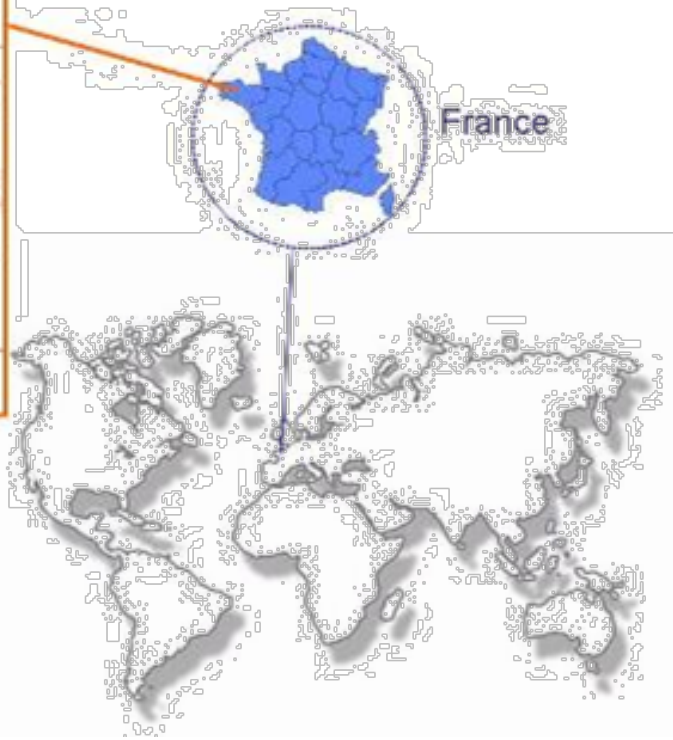
INSIGHTS INTO WAVEFORM PORTABILITY ISSUES OF FM3TR WAVEFORM

Frédéric Le Roy, ENSTA Bretagne
Lahatra Rakotondrainibe, ENSTA Bretagne
Jean-Philippe Delahaye, DGA MI
Ali Mansour, ENSTA Bretagne

At the westernmost tip of the European continent, open to the world





AGENDA

- PORTABILITY CONCERNS IN SDR
- SCA COMPONENT GENERATION
- FM3TR WF ON TWO PLATFORMS
- OBSERVED PORTING LIMITS

PORTABILITY CONCERNS IN SDR

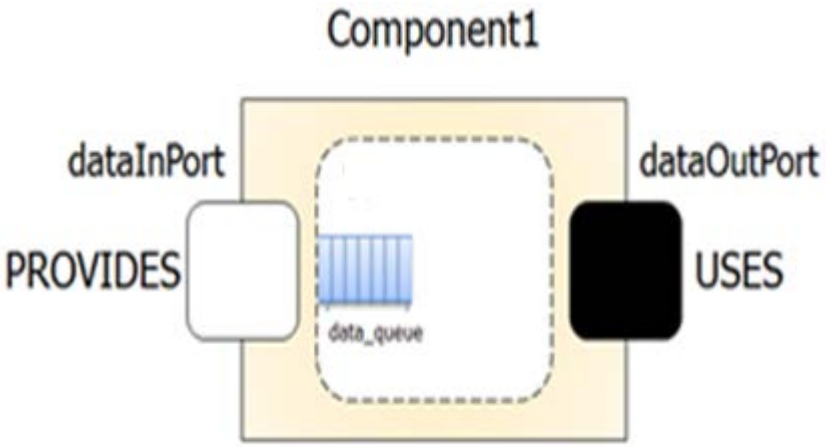
- SCA Hot topic
  - "Wireless Innovation Forum Top 10 Most Wanted Wireless Innovations"
  - JTRS NED, "Waveform Portability Guidelines V1.2.1".
- Multi-aspect problem
  - Variety of resources used by SCA nodes (very high heterogeneity)
    - GPP, DSP, FPGA, NoC, MPSoc, …
  - Middleware used for WF deployment
    - RT management

AGENDA

- PORTABILITY CONCERNS IN SDR
- SCA COMPONENT GENERATION
- FM3TR WF ON TWO PLATFORMS
- OBSERVED PORTING LIMITS

SCA COMPONENT GENERATION

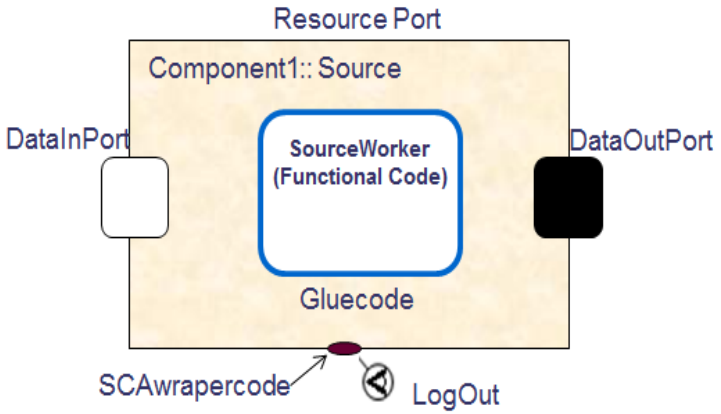
- Example of code generation
  - OSSIE




```
Component1_i::Component1_i(const char *uuid, omni_condition *condition) :
    Resource_impl(uuid), component_running(condition)
{
    dataIn_0 = new standardInterfaces_i::complexShort_p("dataInPort");
    dataOut_0 = new standardInterfaces_i::complexShort_u("dataOutPort");
    start();
}
```

SCA COMPONENT GENERATION

- Example of code generation
  - Zeligsoft CE




```
zceComponent1Servant::zceComponent1Servant( const CORBA::ORB_ptr& orb, const sink_params& execParams ZCE_EXC_ENV_ARG)
{

    ZCE_ASSERT_EXCEPTION_VOID;
    orb_ = CORBA::ORB::_duplicate(orb);

    params_ = execParams;

    state_ = UNINITIALIZED;


    worker_ = new zcesComponent1Worker(execParams ZCE_EXC_ENV_PARAM );
    in_dataInPort = new zceSimpleOctetPacketSinkProvidesPort("dataInPort", SINK_DATAIN, worker_ ZCE_EXC_ENV_PARAM );
    out_dataOut = new zceSimpleOctetPacketSinkUsesPort("dataOutPort" ZCE_EXC_ENV_PARAM );
}
```

AGENDA

- PORTABILITY CONCERNS IN SDR
- SCA COMPONENT GENERATION
- FM3TR WF ON TWO PLATFORMS
- OBSERVED PORTING LIMITS

FM3TR WF ON TWO PLATFORMS

- FM3TR traffic
  - 16 kbps voice traffic, compatible with STANAG 4209 and 4380
  - Instant Text Messaging (ITM)
    - A text message shall not exceed 48180 bytes
- FM3TR specifications
  - DLC (Data Link Control) layer
  - MAC (Medium Access Control) layer
  - PHY (Physical) layer

FM3TR WF ON TWO PLATFORMS

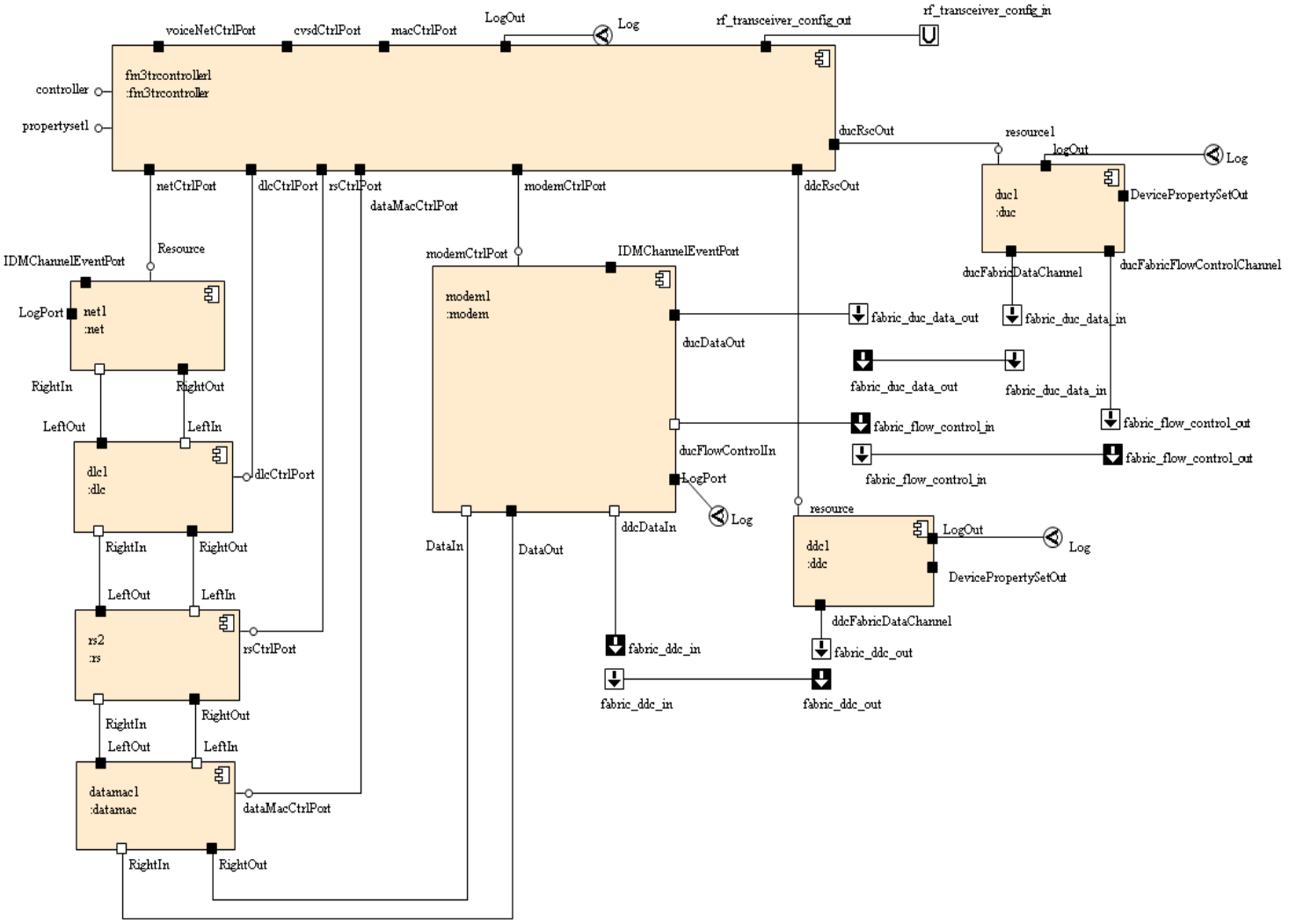
- FM3TR specifications
  - DLC (Data Link Control) :
    - Point-to-Point Protocol (PPP)
  - MAC (Medium Access Control)
    - *p*ersistent Carrier Sense Multiple Access (*p*CSMA)
  - PHY (Physical)
    - Frequency Range : 30-400 MHz (TW1), 225-400 MHz (TW2)
    - Channel spacing : 25 kHz
    - Frequency accuracy : 10^{-6}
    - Modulation type : MSK
    - Hop rate, sets and timings : 250 h/s … 2083.3 h/s
    - Synchronization, Channel Coding

FM3TR WF ON TWO PLATFORMS

- Software architectures (Calit2 sources)
  - Host application
    - Connect and exchange ITM / Voice with “SCA Net Device”
  - CORBA component
    - Data Link Control (DLC) connexion, ARQ
    - Channel coding, RS (Reed Solomon)
    - Medium Access Control (MAC)
    - Net and Modem devices
  - Non-CORBA component (DSP)
    - DSP base band part (MHAL exchange, synchronization, …)
    - FPGA intermediate frequency part (frequency hopping and ADC/DAC management, …)

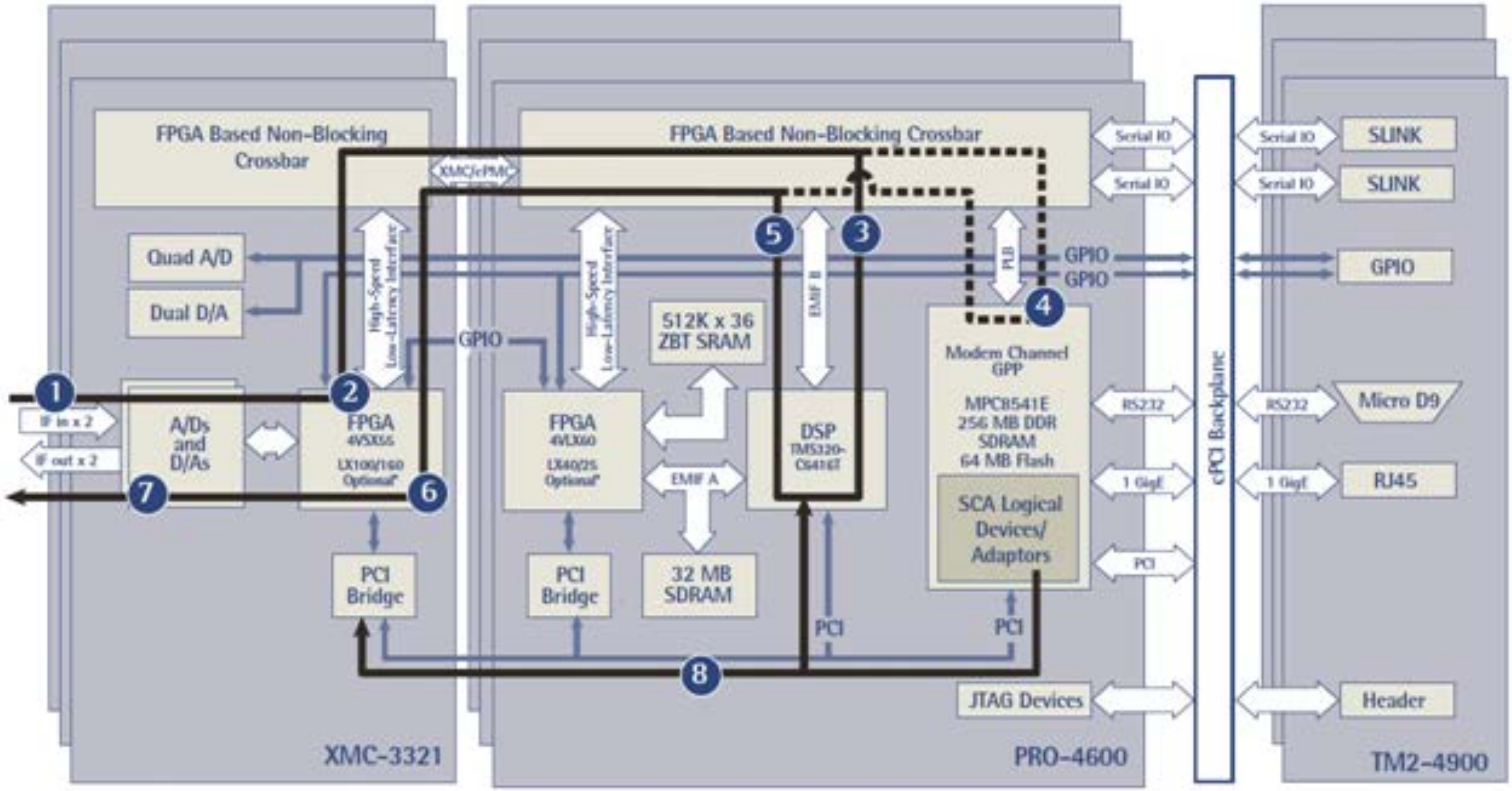
FM3TR WF ON TWO PLATFORMS

- Target waveform SDR-3002



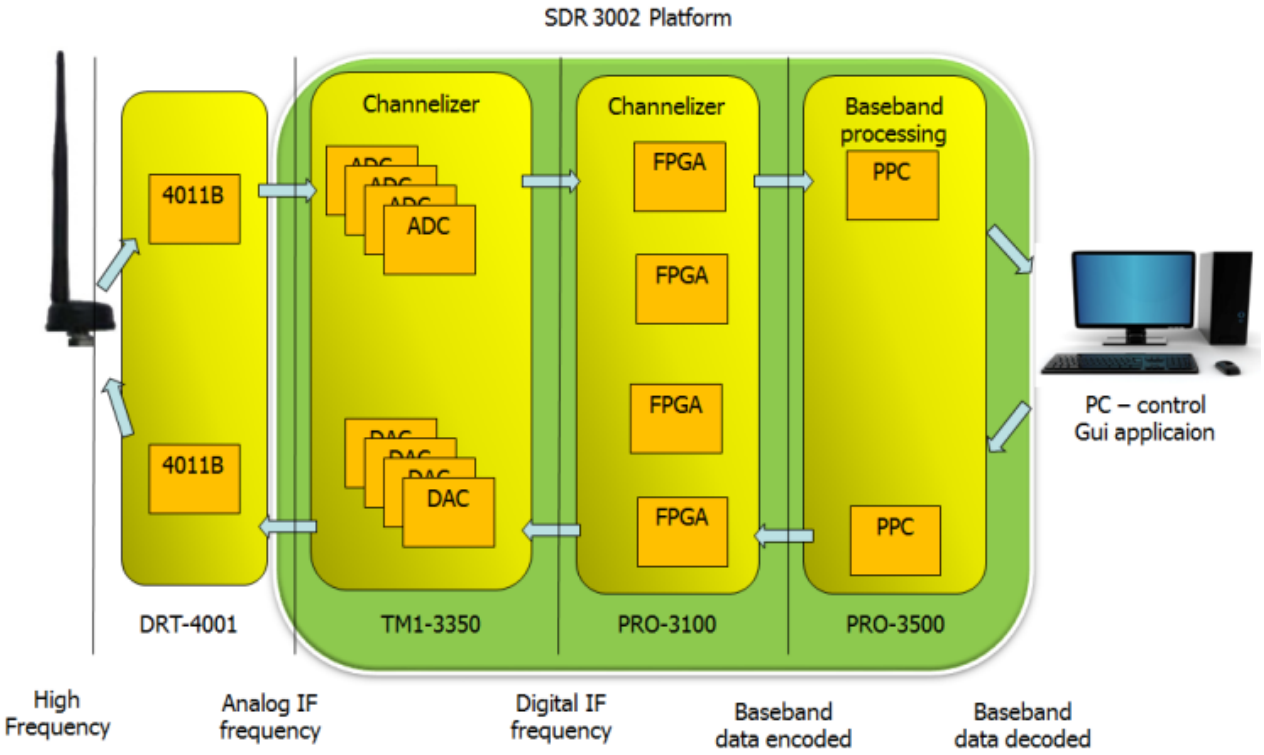
FM3TR WF ON TWO PLATFORMS

- Platform architectures
  - Source platform : SDR-4000



FM3TR WF ON TWO PLATFORMS

- Platform architectures
  - Target platform : based on SDR 3002



FM3TR WF ON TWO PLATFORMS

- Mapping and results

Component	*card*	*Target Circuit*	*OS*
cvsd	PRO3500, EPMC8310	P0, PPC7410	VxWork
datamac	PRO3500, EPMC8310	P0, PPC7410	VxWork
fm3trcontroller	SBC	Pentium	Windows
mac	PRO3500, EPMC8310	P0, PPC7410	VxWork
nspr842_duc	PRO3100	XC2V3000 Virtex-II, SAND 0	VxWork
nspr842_ddc	PRO3100	XC2V3000 Virtex-II, SANN 3	VxWork
rs	PRO3500, EPMC8310	P0, PPC7410	VxWork
net	SBC	Pentium	Windows
voiceNet	SBC	Pentium	Windows
modem_device	PRO3500, EPMC8310	P0, PPC7410	VxWork

AGENDA

- PORTABILITY CONCERNS IN SDR
- SCA COMPONENT GENERATION
- FM3TR WF ON TWO PLATFORMS
- OBSERVED PORTING LIMITS

OBSERVED PORTING LIMITS

- Development tool limitations
  - Component interface

```
interface IoPacket {
  Oneway void pushPacket
 (in CF:OctetSequence Payload);};
-----------------------------------------------------
interface SimpleOctetPacketSink {
 void pushPacket
 (in NullControl unusedControl,
 in CF:OctetSequence Payload)
 raises (PushPacketFailure);};
-----------------------------------------------------
interface OctetStream : PayloadStatus {
  void pushPacket
 (in StreamControlType control,
 in JTRS::OctetSequence payload)
 raises( UnableToComplete );};
```

OBSERVED PORTING LIMITS

- Development tool limitations
  - SCA component generation process
    - Client / Server Development
    - Component-Based Software Development

OBSERVED PORTING LIMITS

- CORBA and MOC limitations
  - CORBA messaging
    - “one-way” : the client thread used to make the invocation is blocked until the return message is delivered.
      - SYNC_NONE, SYNC_WITH_TRANSPORT, SYNC_WITH_SERVER, SYNC_WITH_TARGET
    - “two-way”: the execution thread of the client resumes before the member function is invoked on the server side.
  - OS
    - dmtkThread vs pthread

OBSERVED PORTING LIMITS

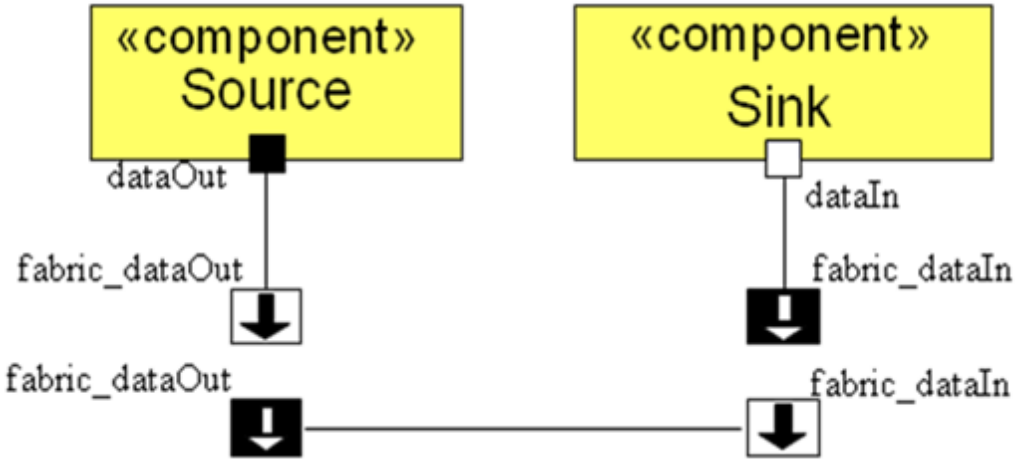
- Illustration

```
/*-------BEGIN CALIT ORIGINAL CODE-------------
pthread_t _tx_thread;
static void* tx_task(void* data);
deque<PortTypes::ShortSequence_var> _tx_queue;
pthread_mutex_t _tx_queue_lock;
sem_t *_tx_sem1;
sem_t *_tx_sem2;
unsigned int _ack_recvd;
int _retry_interval;
int _retry_max;
unsigned char _address;
unsigned char _sequence;
unsigned char _neigh_seqs[256];
int _state;
Boolean _started;
------END CALIT ORIGINAL CODE-------------*/
```

```
dmtkThread *pStartThread_;
static void zcedlcWorker::dlc_task( void* thisPtr );
void zcedlcWorker::tx_task(void);
deque<PortTypes::ShortSequence_var> _tx_queue;

SEM_ID Lock_;// DMTK_OS_VXWORKS

sem_t *tx_sem1_;
sem_t *tx_sem2_;
unsigned int ack_recvd_;
int retry_interval_;
int retry_max_;
unsigned char address_;
unsigned char sequence_;
unsigned char neigh_seqs_[256];
int state_;
CORBA::Boolean started_;
int destroyed_;

CORBA::UShort numretrans_;
CORBA::UShort radioid_;
CORBA::UShort retransintvl_;
```

OBSERVED PORTING LIMITS

- Platform limitations



  - Specific parameters injected in WF model by Modeling tools and platforms specific capabilities.

CONCLUSION

- Our experiment conclusion
  - Even though the two platforms are quite similar, the portability of the code is not an easy task to do (SDR3-002 vs. SDR-4000)
  - Code portability depend on :
    - Middleware configuration
    - SCA component generation (SCA development tools).
    - OS
    - Fast link interconnection use

CONCLUSION

- How to enhance waveform portability
  - A WF must be defined at multiples levels : ESSOR, MDE
    - “Base” waveform can be simulated (PIM)
      - At multiple abstract levels (UML Modeler/Simulator, Opnet, Simulink)
    - “Target” Waveform is deployed on SCA platform (PSM)
      - FM3TR Physical layer involves FPGA and DSP
        - » Devices are difficult to test !
  - Multi-facet problem : standards API
    - Waveform developers
    - Device and platform developers
    - Framework developers

CONCLUSION

- How to enhance waveform portability
  - Use SCA development tools with full support of SCA requirements
  - Device mustn't be recoded by the WF developer
    - Ex : recoding a NET device that doesn’t support TCP !
  - Golden code should be :
    - located in a worker class
    - generated from “Base” component and SCA component Meta-model (from SCA component generator tool).
      - Open SCA Meta-model and M2M transformation.

ENSTA Bretagne

THANK YOU !

Question ?

- What can SCA 4.1 do better ?
  - Where are most important portability difficulties ?
    - In physical layers on DSP and FPGA target
  - SCA 4.1 gives partial solutions with Lw AEP
  - ESSOR offers response for DSP and FPGA