An Approach for solving Real-time and Synchronization issues in heterogeneous Multi-Processor Software Defined Systems

Peter Troll, Rohde & Schwarz

WInnComm Europe, 15 May 2019, Berlin

ROHDE&SCHWARZ

Outline

- **The SDR Standards Ecosystem**
  - A selective Review on Standards, Concepts, Terminology
- **Focus: Air-interface Synchronization** - The approach explained step by step
  - Starting with WInnF Tranceiver Facility
  - How to combine existing standards: WInnF Transceiver & JTRS Timing Service
  - How to exploit FPGA computational element type real-time capabilities
    - RT-capable communication FPGA ⇔ GPP|DSP
- **An exemplary SDR system** – The approach in a nutshell
  - Platform/HW Architecture, Application Architecture, Synchronization Principles
- **Summary**
  - Key technology ideas behind the approach
  - SCA vs. non-SCA environments
  - General considerations and strategy review

Let's get started …

… by a sort of trivia game:

9,192,631,770 **?**

Any ideas?

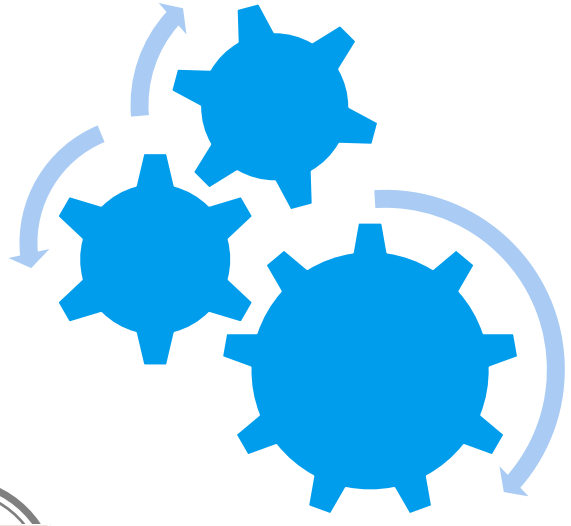
Caesium Atomuhr „CS 4“ [Wikipedia]

"... periods of the radiation corresponding to the transition between the two hyperfine levels of the ground state of the caesium 133 atom" [BIPM]

... are **one second** !

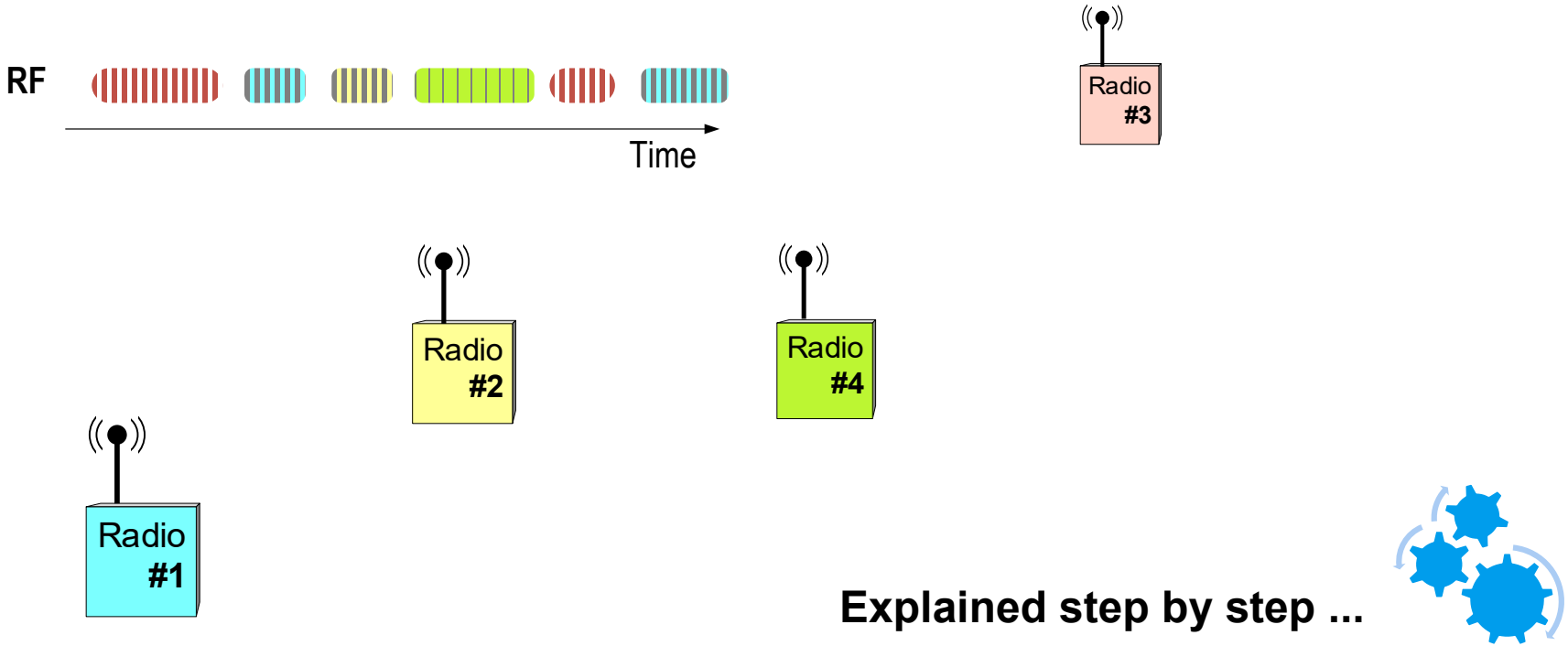
A short Review on Standards, Concepts and Terminology

- **SCA** - Software Communications Architecture
  - **AEPs** - Application Environment Profiles
- IEEE/OMG **POSIX** Real-time Support
  - Clocks and Timers Interfaces
- JTRS **Timing Service** API
  - The *Terminal Time* Concept
- JTRS **MOCB** API - MHAL on Chip Bus
  (MHAL = Modem Hardware Abstraction Layer)
  - Event Interface
- WInnF **Transceiver Facility** PIM Specification
  - Monotonic Clock Absolute Time Controlled Transceivers



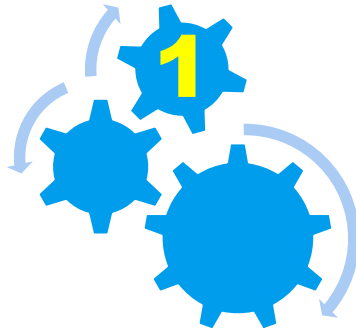
An Approach for Solving Real-time and Sync Issues in SDS

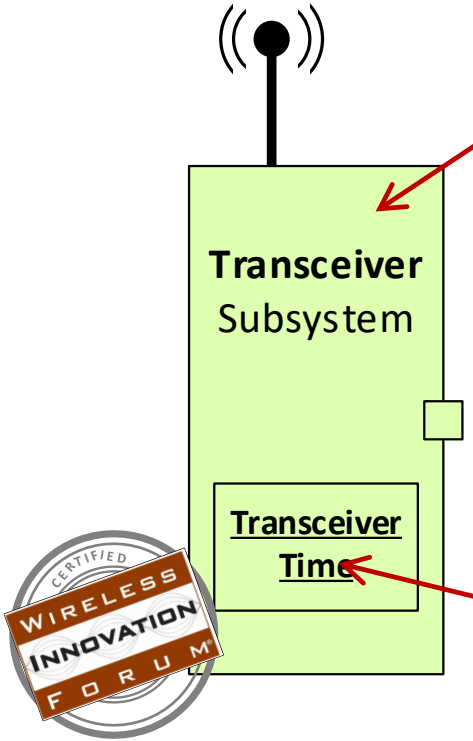
Focus: SDR‘s Air-Interface Synchronization



Explained step by step ...

Starting with WInnF Transceiver Facility



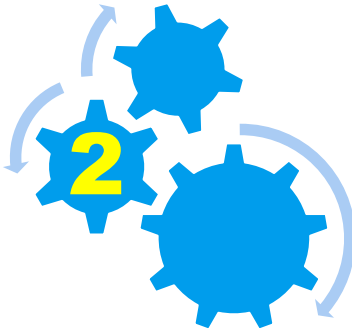


1| WInnF **Transceiver Facility**
PIM Specification Version 2.0.0
compliant subsystem and abstraction

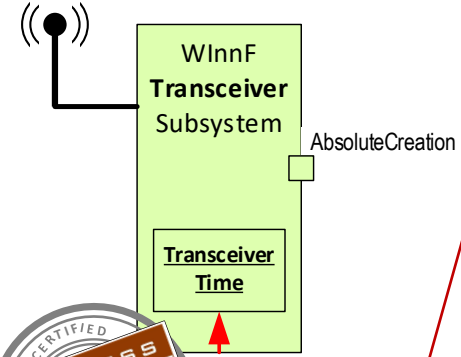
3| ... and corresponding
AbsoluteCreation Interface

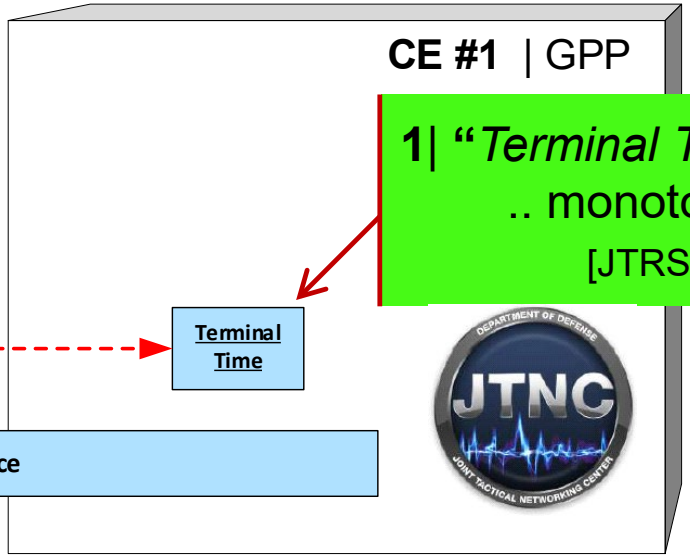
2| Monotonic clock based absolute
Transceiver Time (Elapsed sec/nsec) ...

Teaming up WInnF Transceiver Facility & JTRS Timing Service



2| The ***Timing Service*** **synchronizes the Transceiver with *Terminal Time***

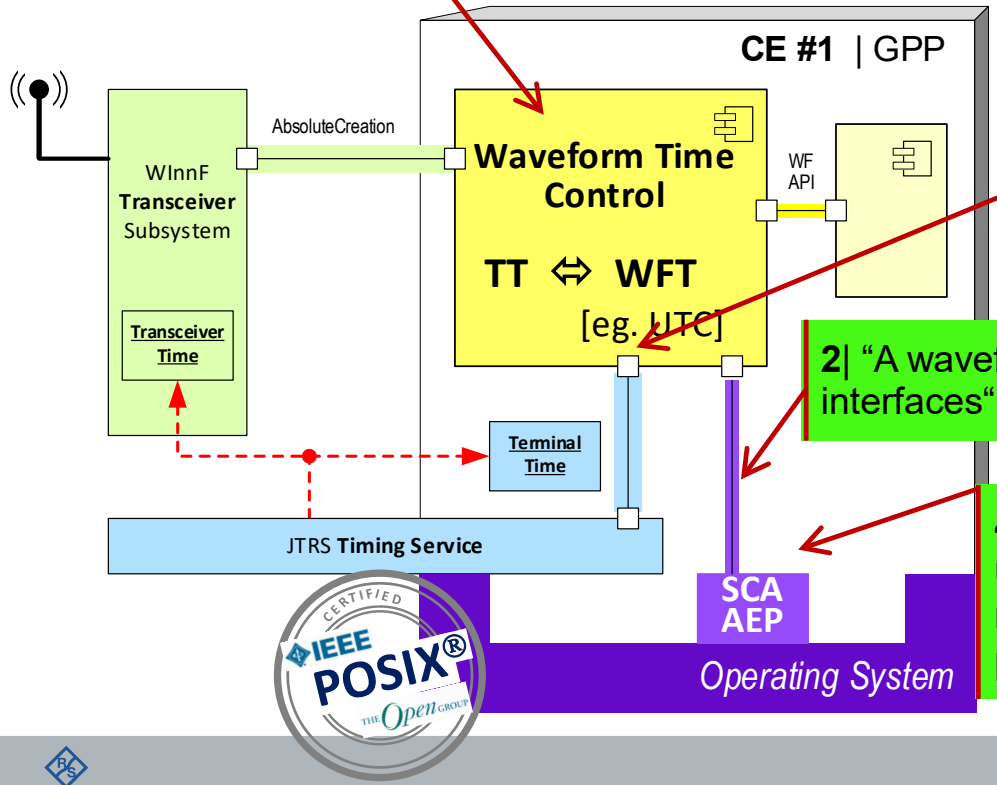




1| "*Terminal Time* is …
.. monotonic increasing"
[JTRS Timing Service API]

In SCA / POSIX Environments

1| Applications implementing a WF with requirements to precise **air-interface synchronization** will translate between the specific waveform time format and monotonic Transceiver/Terminal Time.



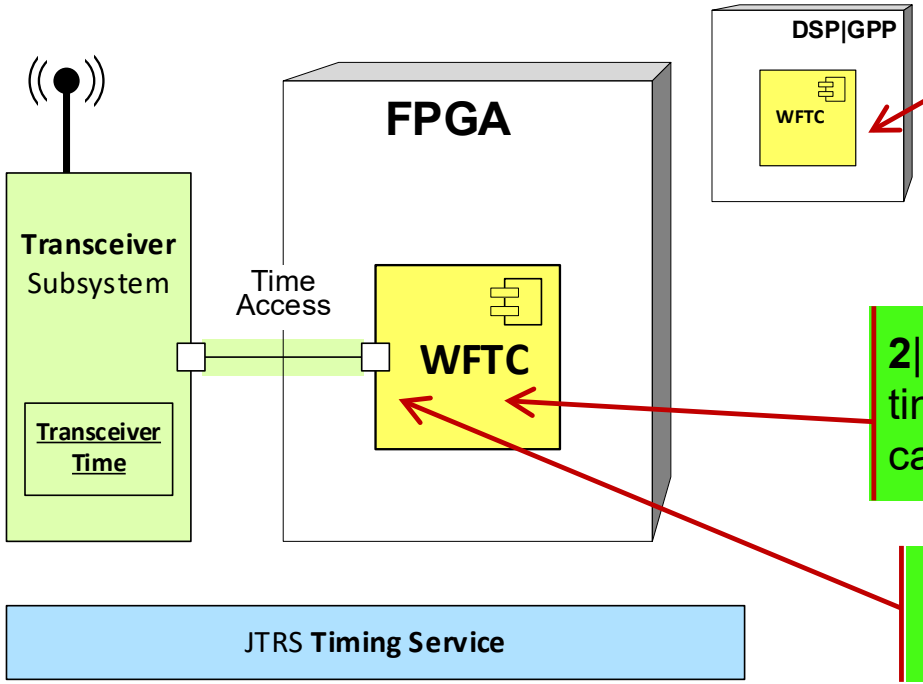
3| Application retrieves *System Time* (UTC) or previously stored *WFTime* from Timing Service API.

2| "A waveform retrieves *Terminal Time* via the POSIX time interfaces" → `clock_gettime(CLOCK_MONOTONIC)` [JTRS TS]

4| POSIX Clocks and Timers RT Support provides us with high-resolution timer functions for implementing time management operations such as periodic activations, short duration timeouts, etc.

Non-SCA-Environments

Exploit FPGA Real-time capabilities



3| But:
- Need SW support!
- Need UTC
- What about timers?

➔ xx**P/FPGA** Collaboration

2| Allows the Application Designer implementing waveform specific time management capabilities to his WF components

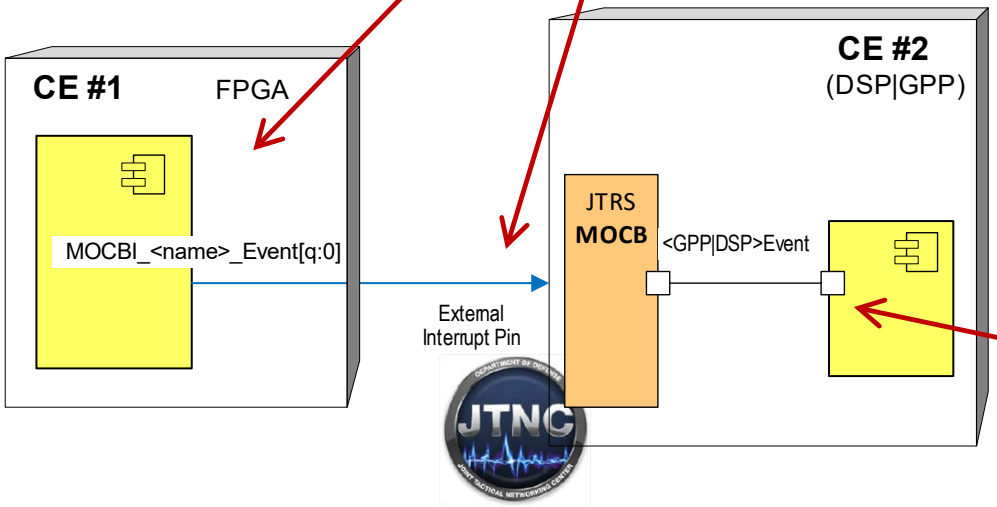
1| Get *Terminal / Transceiver Time* from the Transceiver‘s *TimeAccess* Interface

FPGA⇔GPP|DSP Real-time capable communication

JTRS MHAL on Chip Bus (MOCB) Event

1| MOCB FPGA API Event signals

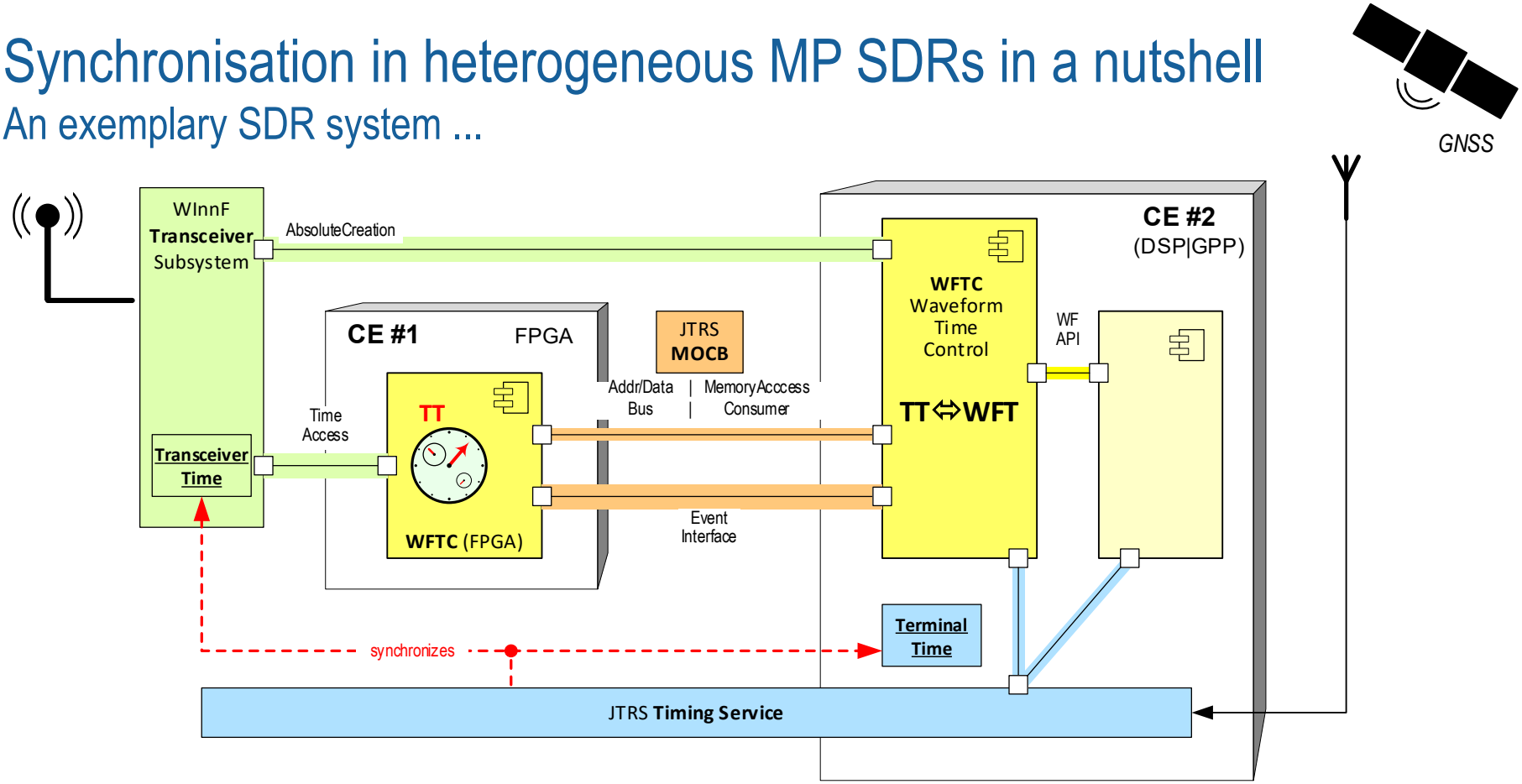
2| Typically connected to processors hardware interrupt capable pins



3| MOCB GPP|DSP API **Event** interface based on SEMAPHORE concept (→ POSIX)

Synchronisation in heterogeneous MP SDRs in a nutshell

An exemplary SDR system ...



... Fully Standard compliant Host Environment

- **WInnF Transceiver Facility V2**
  - Monotonic Clock Absolute Time Controlled Transceiver
  - Transceiver Time synchronized to Terminal Time
  - TimeAccess Interface (@ FPGA)
- **JTRS Timing Service API**
  - Synchronizes Transceiver with Terminal Time
- **JTRS MHAL on Chip Bus (MOCB) API**
  - JTRS MOCB Event signalling mechanism for FPGA to GPP|DSP interconnect

... The approach from application point of view

- Exploit **FPGA CE**‘s technology **hard real-time** capabilities
- Implement waveform specific synchronization needs
  - Get Transceiver/Terminal Time Awareness into the waveform
  - Establish and maintain relationship Terminal Time ⇔ Waveform Time
- Application Architecture and Design:
  - Consider deployment best suitable for your specific needs
  - Consider proper waveform internal API
  - Consider application portability

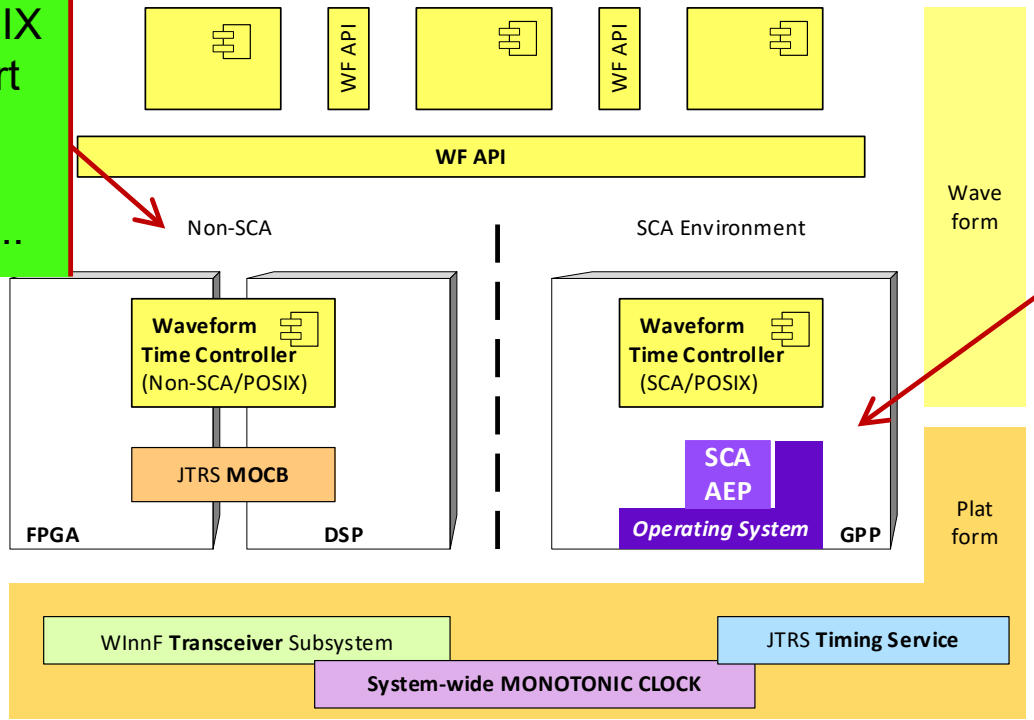
Summary - Key technological ideas

- Establish **System-wide Monotonic Clock**
  - Converge/combine **WInnF Transceiver Facility** and **JTRS Timing Service** Concepts (*Transceiver Time* = *Terminal Time*)
- Thourougly consider **different CE type‘s real-time capabilities**
  - Apply concepts/standards best suitable e.g. JTRS MOCB Event signalling mechanism for FPGA to GPP|DSP interconnect
  - Allow waveform to take maximum advantage of CE‘s RT capabilities
- In general: Adopt **established RT concepts** (Particularly **POSIX**)
  - Apply to „non-SCA“ Environments

Summary - SCA vs. non-SCA

1| It's just about SCA AEP / POSIX (and not about SCA CF functionality)

2| No POSIX RT support necessary with the approach ...



3| .. but based on a common set of generic concets (like monotonic clock).

Finally a result from SCA and JTRS API adopting POSIX concepts!

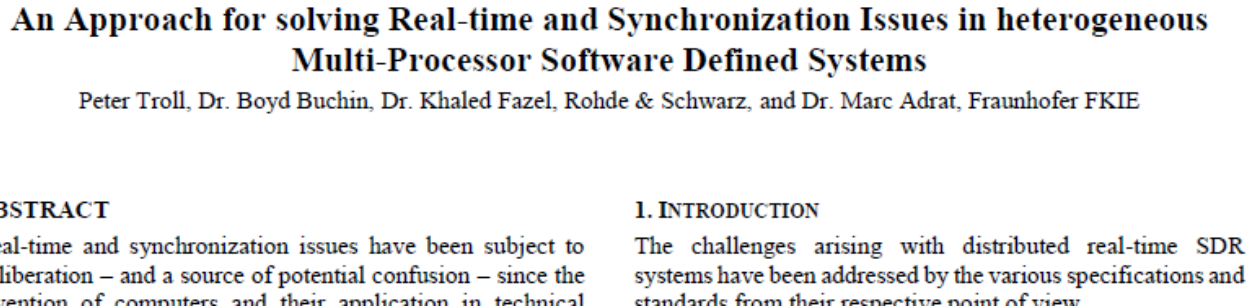
Summary - Overall Strategy für distributed SDS
Platform Architecture, Application Design & Portability

- Keep host environment lean and simple
- Leverage existing concepts and standards
- Provide waveform-agnostic abstraction of functionalities
  - No assumptions on details what a waveform will need and how an application will implement
  - Implement waveform specifics in the application
- Consider application architecture
  - Maximize percentage of components likely to be ported with little or no expense

An Approach for solving Real-time and Synchronization issues in heterogeneous Multi-Processor Software Defined Systems

THANK YOU.

Paper will be available with proceedings ...



An Approach for solving Real-time and Synchronization Issues in heterogeneous Multi-Processor Software Defined Systems

Peter Troll, Dr. Boyd Buchin, Dr. Khaled Fazel, Rohde & Schwarz, and Dr. Marc Adrat, Fraunhofer FKIE

ABSTRACT

Real-time and synchronization issues have been subject to deliberation – and a source of potential confusion – since the invention of computers and their application in technical

1. INTRODUCTION

The challenges arising with distributed real-time SDR systems have been addressed by the various specifications and standards from their respective point of view