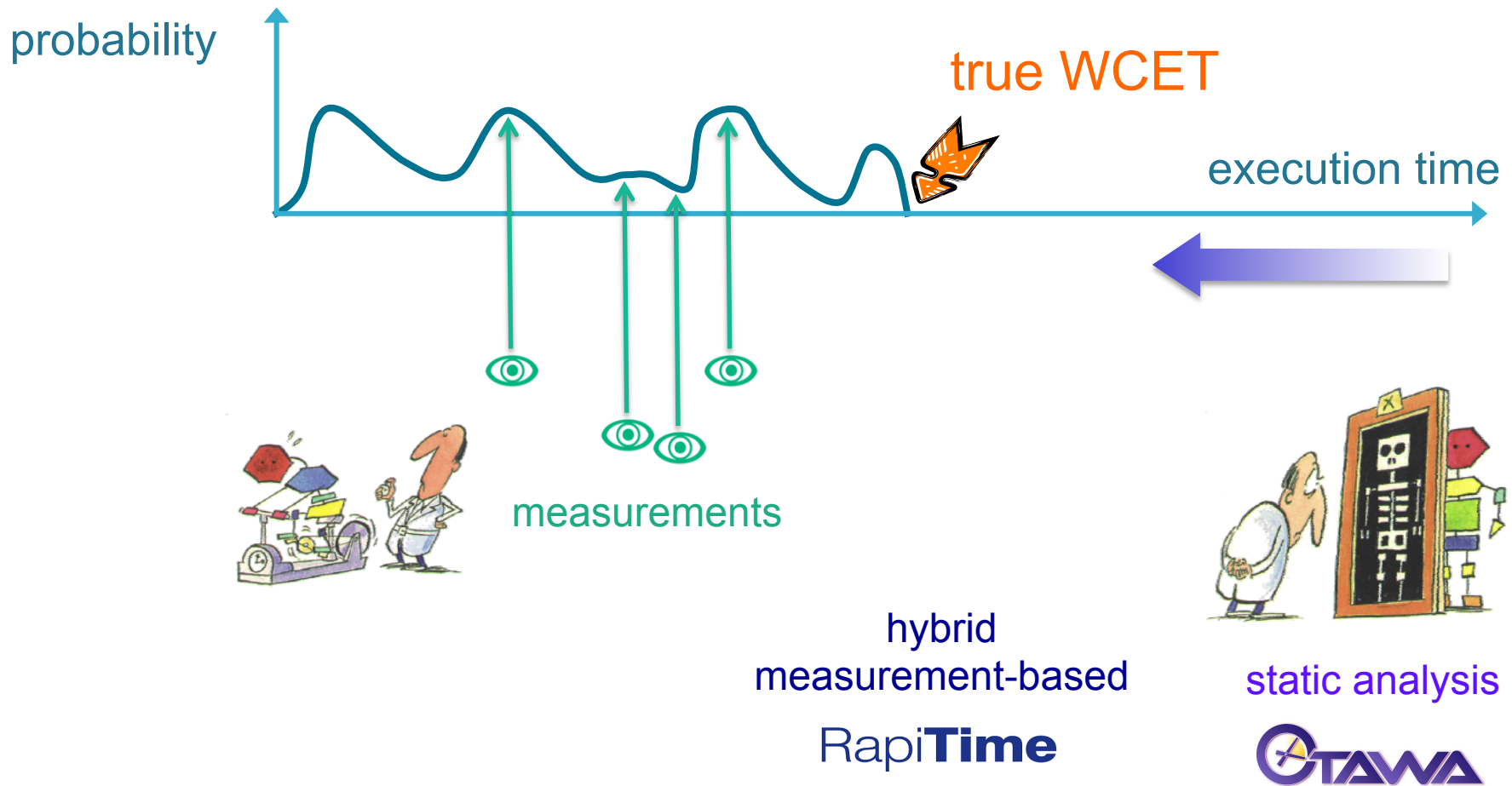
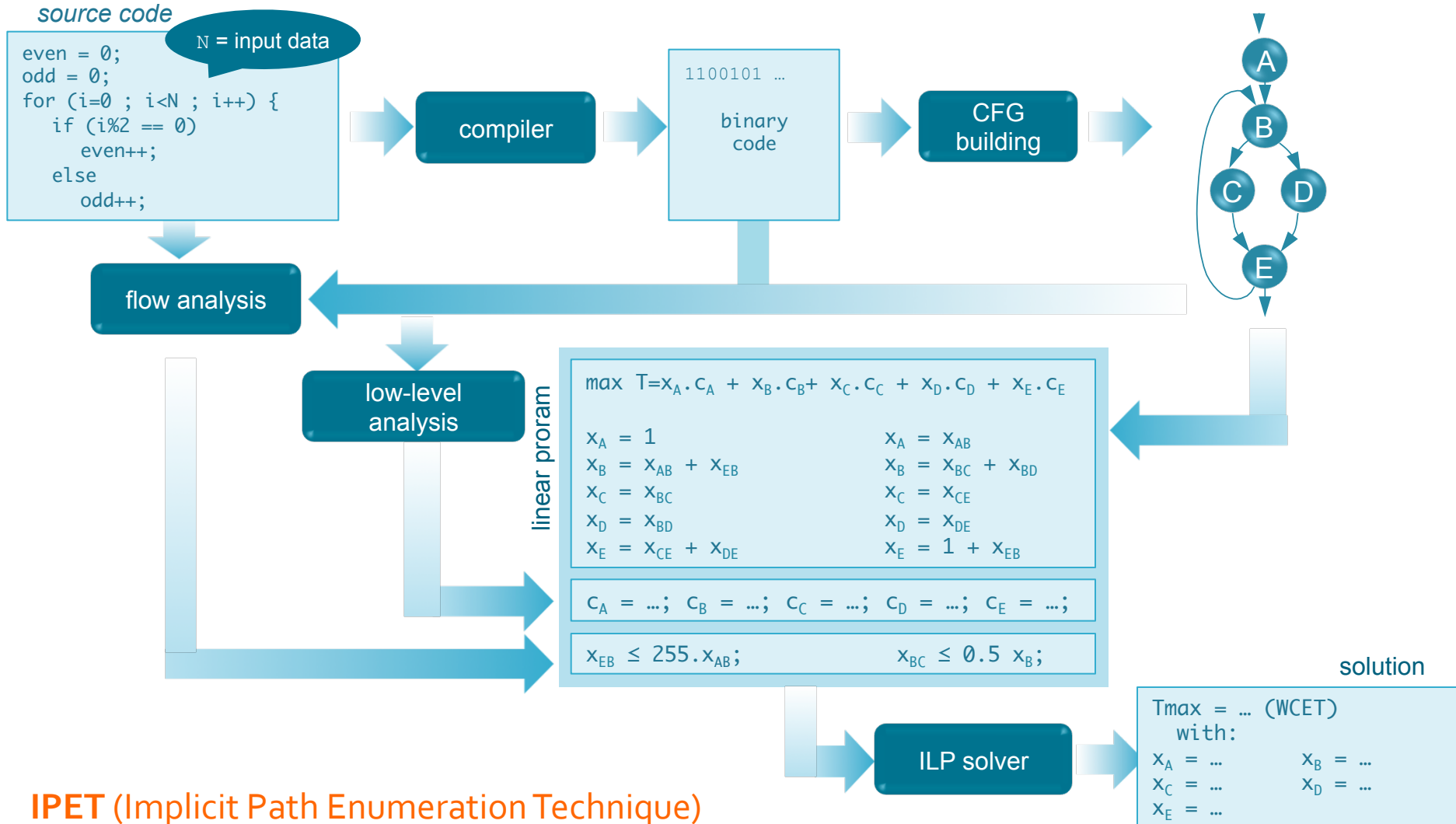


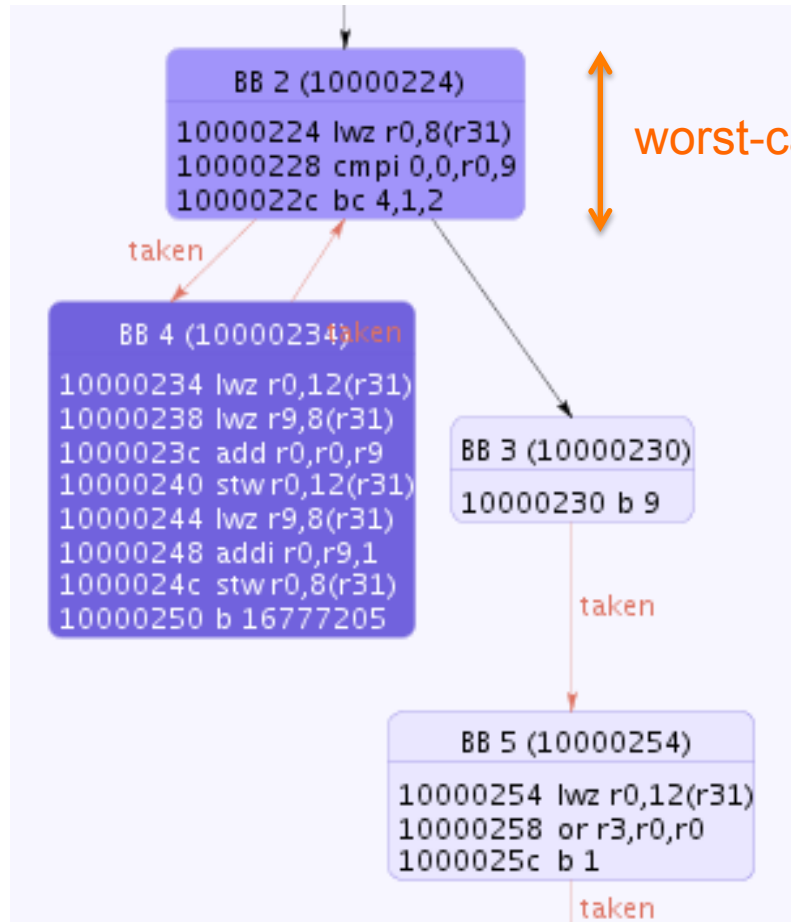
parMERASA

WCET Analysis Tools

Haluk Ozaktas (Univ. of Toulouse)







worst-case cost of a basic block?

instruction fetch latencies?
data load/store latencies?

1st challenge: multicore architecture

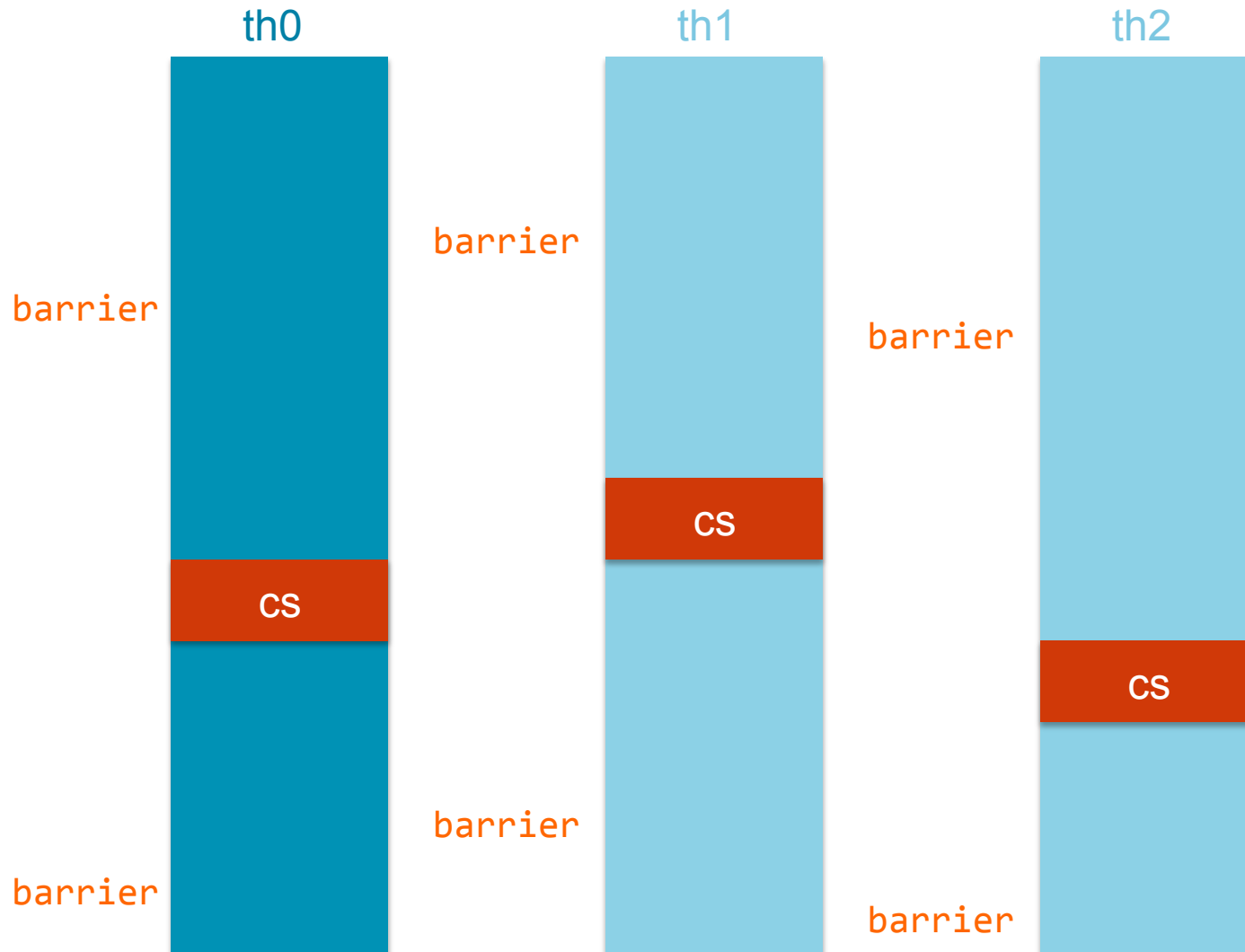
conflicts to shared resources?

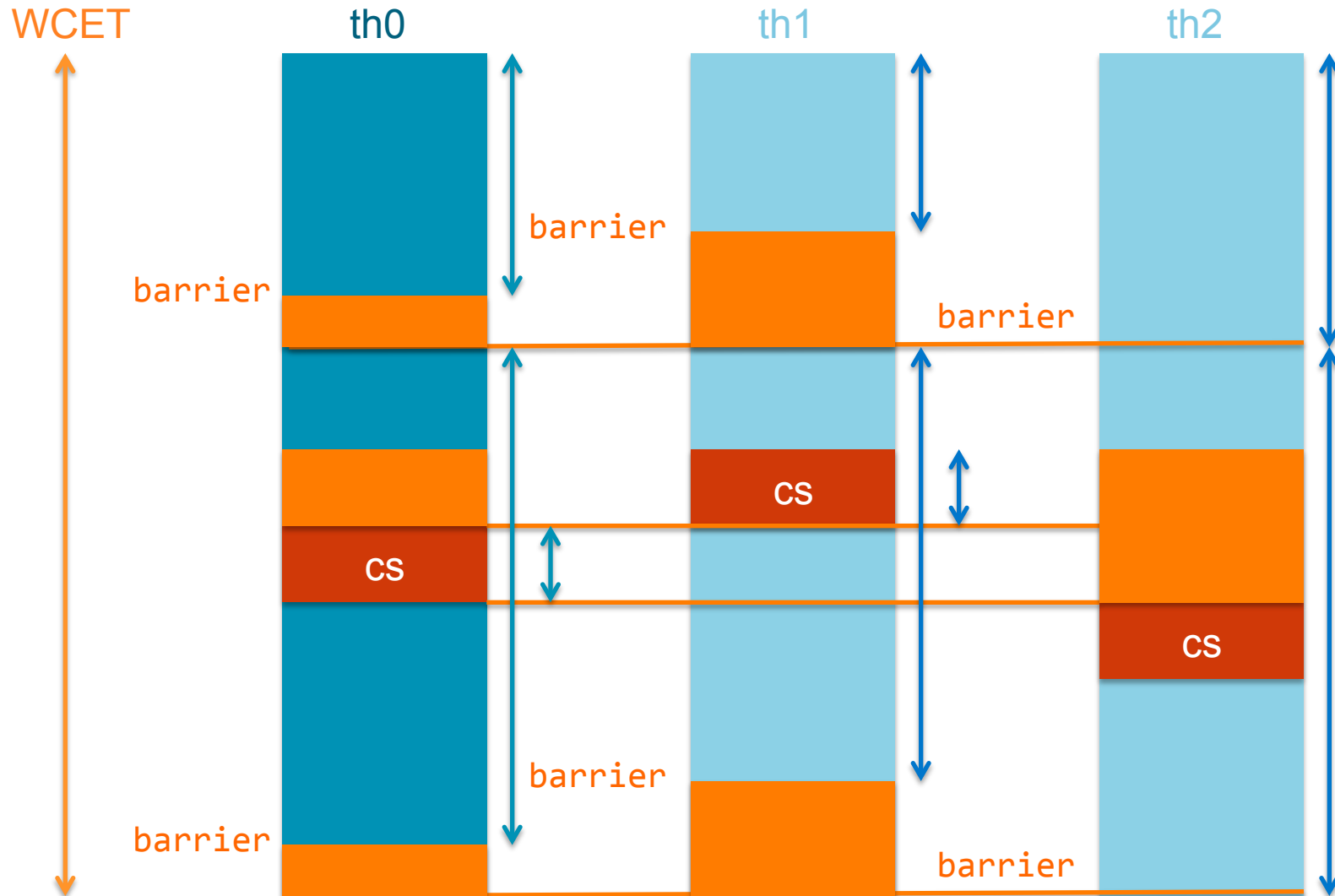
⇒ **time-predictable** architecture
(*known upper bounds for latencies*)

2nd challenge: parallel programming

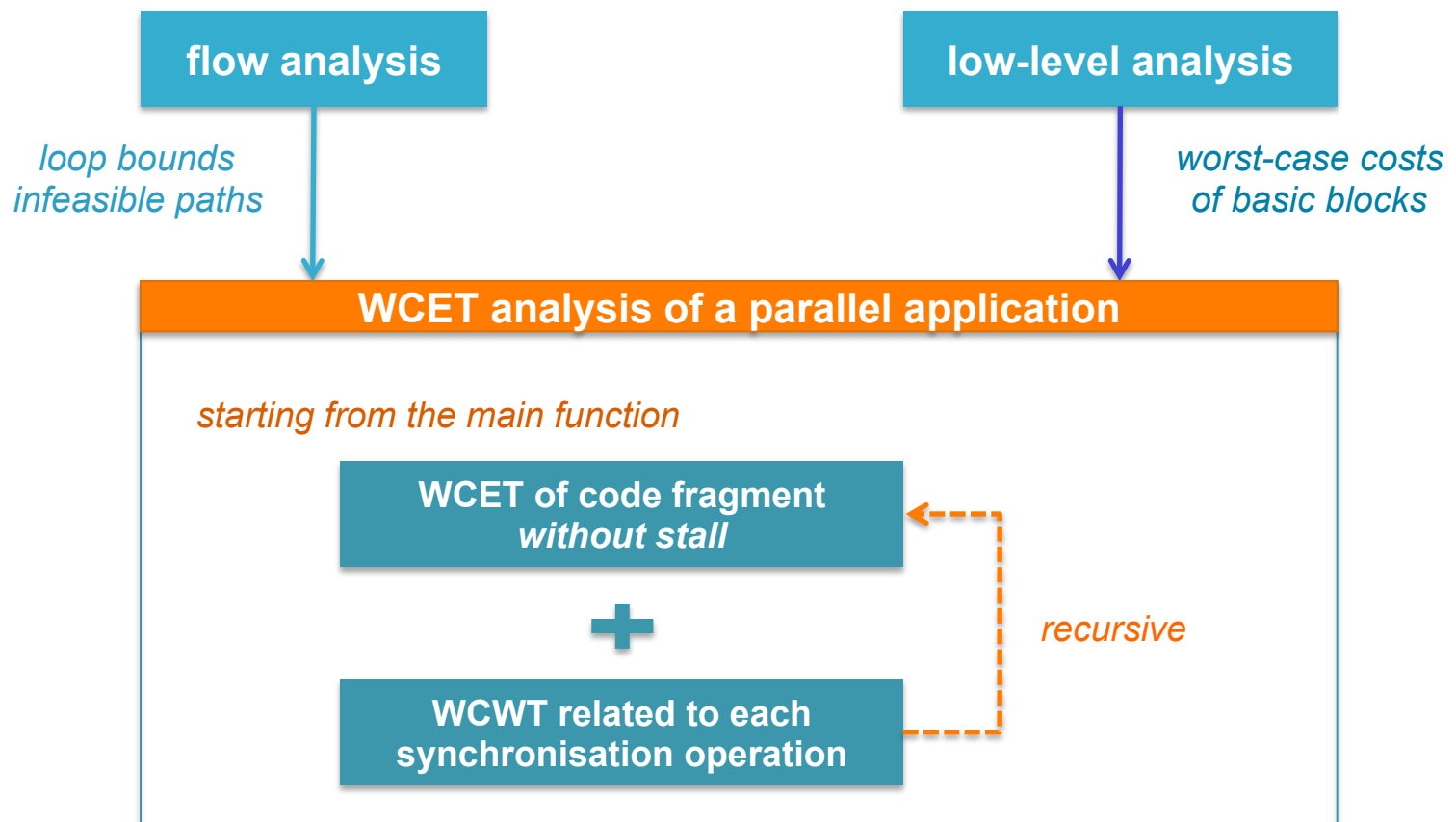
inter-thread synchronisations generate
waiting times

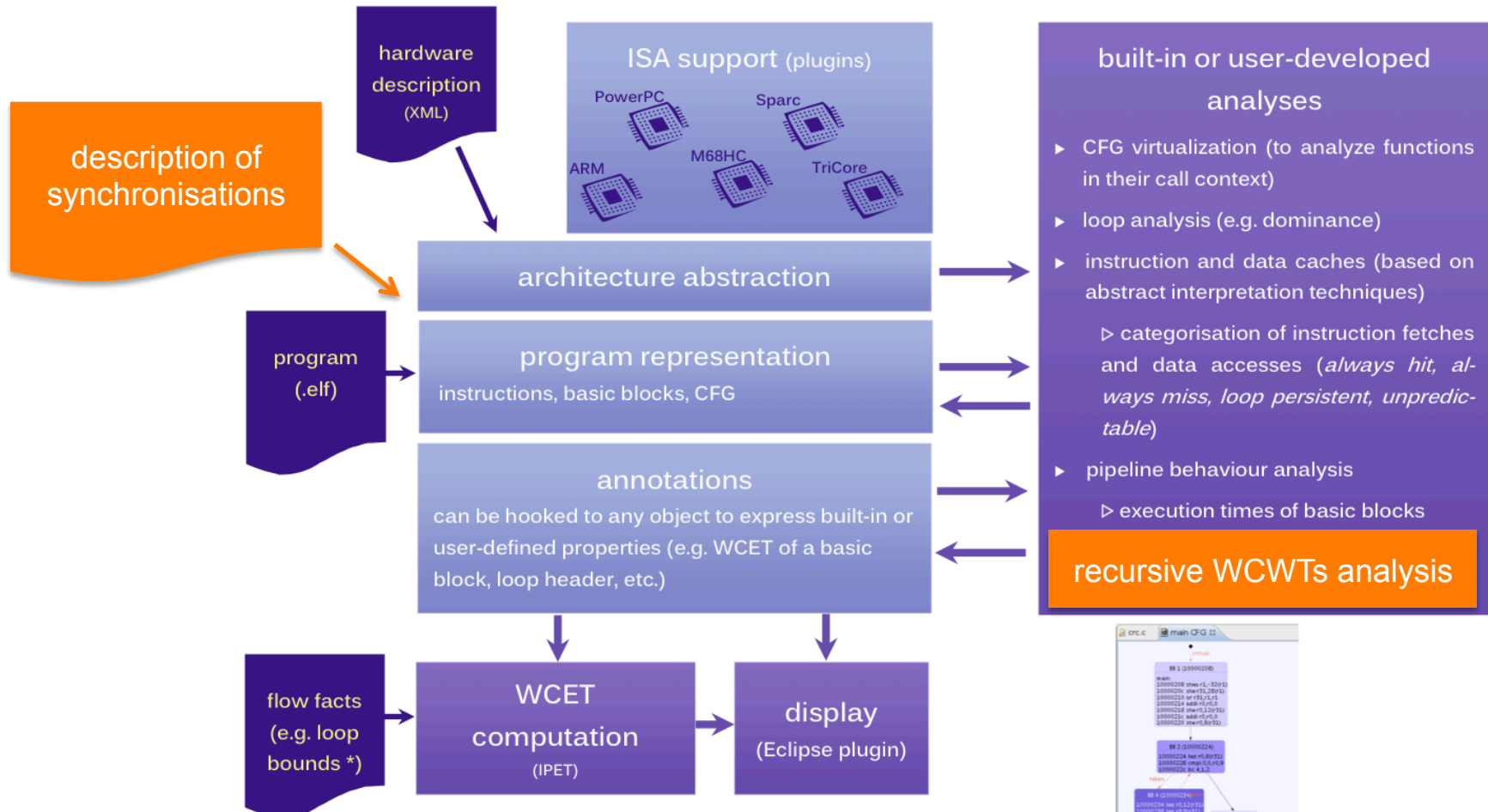
⇒ need to bound them!



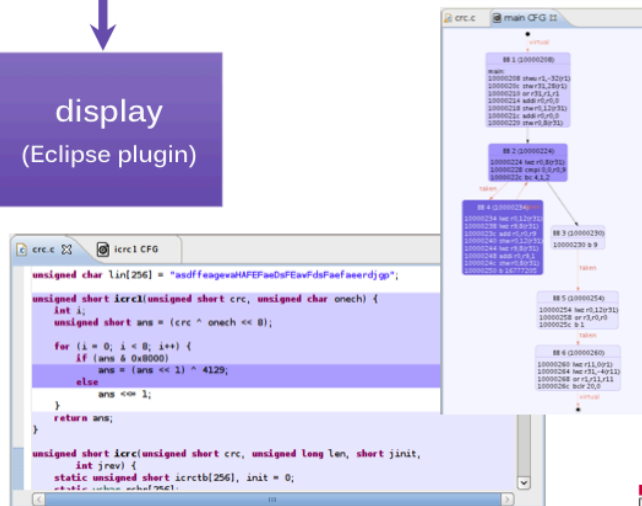


■ Computing WCWTs





* Our oRange tool determines loop bounds from source code analysis



■ Current state

- General procedure defined
- Annotation language specified
- Preliminary version implemented and used on toy examples

■ Future work

- Refined analysis of synchronisation primitives
- Assistance to code annotation
- Visualisation

parMERASA

WCET Analysis Tools

Haluk Ozaktas (Univ. of Toulouse)

Verification and Profiling Tools in parMERASA

Dr Ian Broster
Rapita Systems Ltd

parMERASA
Multi-Core Execution of *Parallelised* Hard Real-Time
Applications Supporting Analysability

(Context: Verification of Parallel, critical systems software for Multicore)

Software Design

- Understanding legacy software and dependencies
- Concurrency bugs, diagnostics
- Complexity of communication

Timing verification

- Hardware: cores interact: buses, shared memory, cache coherency, ...
- Software: parallelized applications are complex – synchronization and waiting times

Testing

- Visualization & Comprehension
- Coverage/on-target testing
- Repeatability/testability



Extensions of RVS technology

Design
Support

- Dependencies in legacy software
- Communication profiling

In-depth
tracing

- Visualization techniques
- Graphical view

Timing analysis

- Rapi**Time** for Multicore
- Worst case execution times
- Multi-core profiling

- First stage: derive a master set of tool requirements

parMERASA
BAUER Maschinen GmbH
Requirements Specification

parMERASA
Honeywell
Requirements Specification

Requirements for Profiling tools

1. The profiling tool shall visualize the inter-module communication.
2. It shall be possible to select all or a subset of global variables for analysis.
3. The profiling tool shall collect the points in time when a task starts or stops to read or write a global variable.
4. The profiling tool shall visualize the time spent for each module.

parMERASA
Tool Requirements

The parMERASA project requires the development of tools to support the development of parallelised software applications, and to verify the timing and other characteristics of parallel software running on the parMERASA architecture.

To ensure that the tools developed during the project meet the aims of the project and the needs of the participants, tool development is guided by requirements. The master set of requirements is derived from a number of sources, including:

- The requirements from each of the industrial partners, and
- The parMERASA DoW, the objectives of which specifies a number of tools to be developed as part of the project.

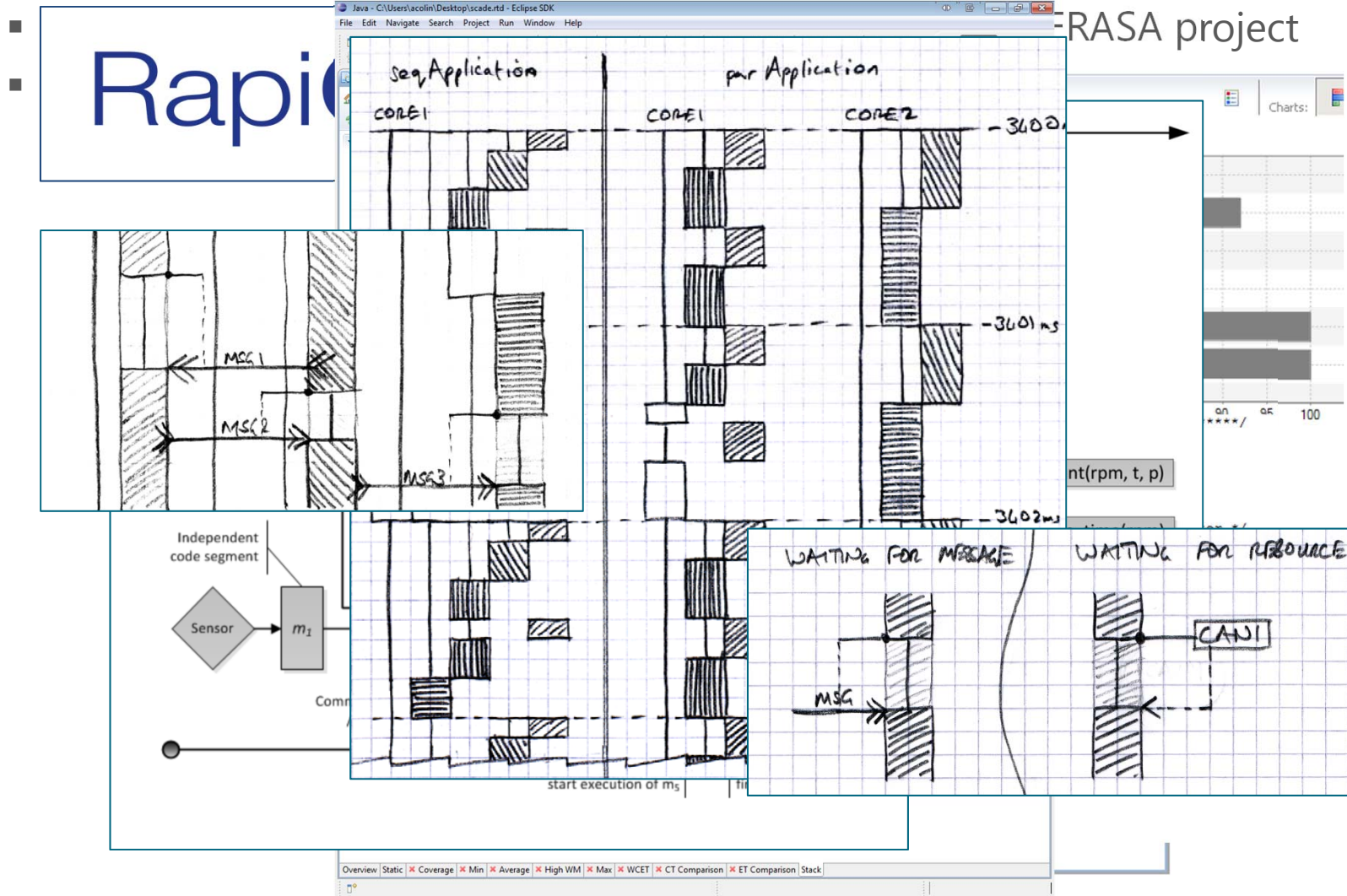
DoW objectives:

The tool-related objectives specified in the Description of Work give rise to the following set of project-based requirements:

- P1: Develop software support for the WCET analysis of parallel programs. This will include innovative code analysis techniques and support for user-defined annotations (O3.2).
- P2: Validate the hardware architectural choices of parMERASA architecture with respect to timing analysability and to worst-case performance. The OFAWA and RapTime tools will be extended to support explored hardware features and the parMERASA architecture (O3.3).
- P3: Develop WCET analysis tool for parallel programs running on parMERASA architecture. This includes the design of frameworks to automatically analyse parallel hard real-time programs. Techniques for determining the pessimism of WCET estimates will be provided (O3.4).
- P4: Develop code coverage and memory analysis tools. They will assist in performance analysis and optimisation of parallel programs (O3.5).
- P5: Develop program profiling tools and visualisation of performance bottlenecks of parallel applications (O3.6).

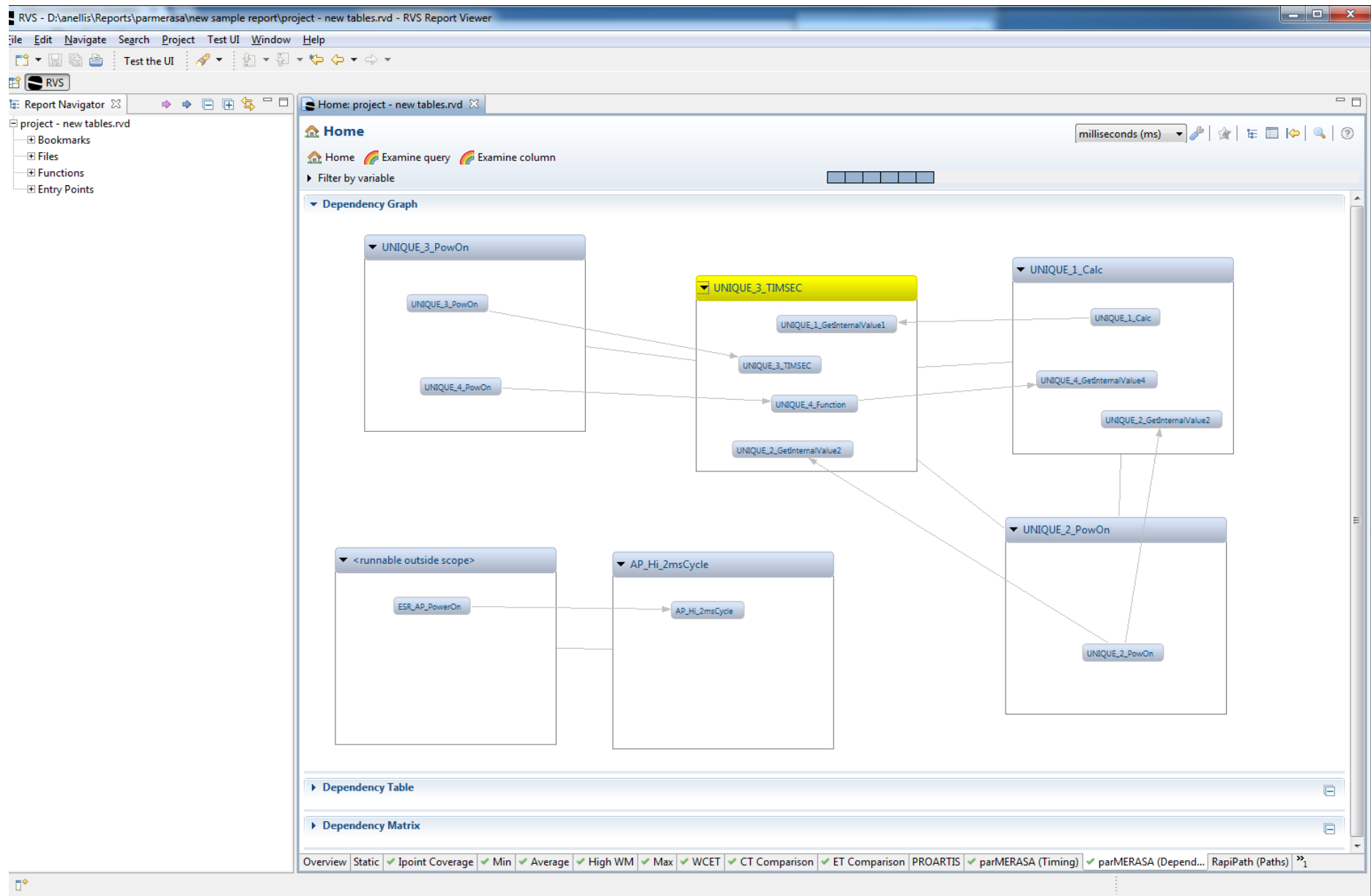
Partner requirements:

The tool requirements for each of the industrial partners have been gathered from items of documentation included in the parMERASA Subversion repository. These requirements will be reviewed in conjunction with the industrial partners to ensure that they accurately reflect the needs



5: Visualization and profiling tool for parallel programs

- To parallelize old/legacy software – where do you start?
 - Minimize the communication
 - Identify independent parts
- Use trace/profiling information to support parallelization



RapiTime Multicore

Automated timing measurements for multicore

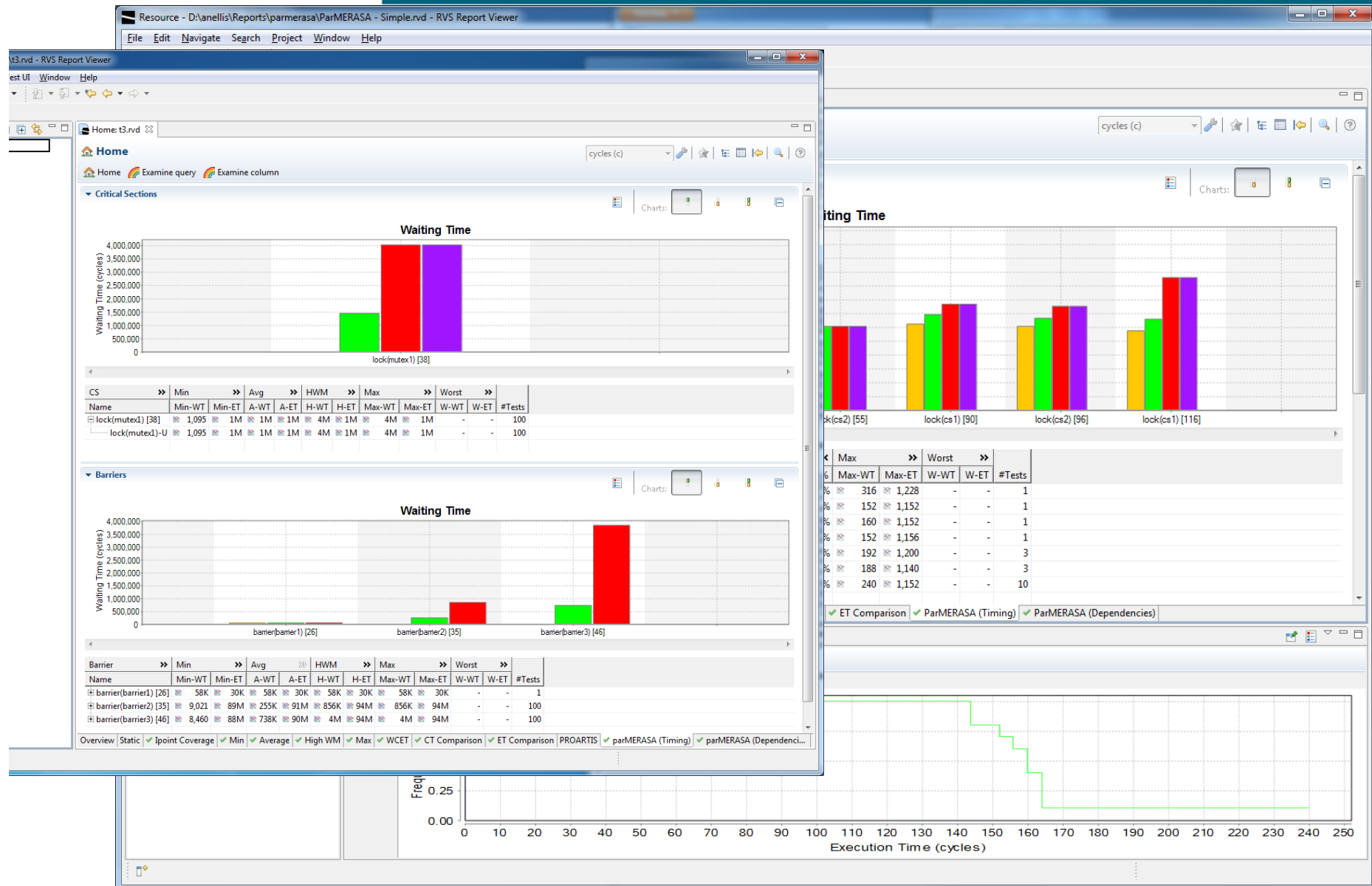
- Blocking, waiting, synchronization times
- Analyse parallel activities simultaneously

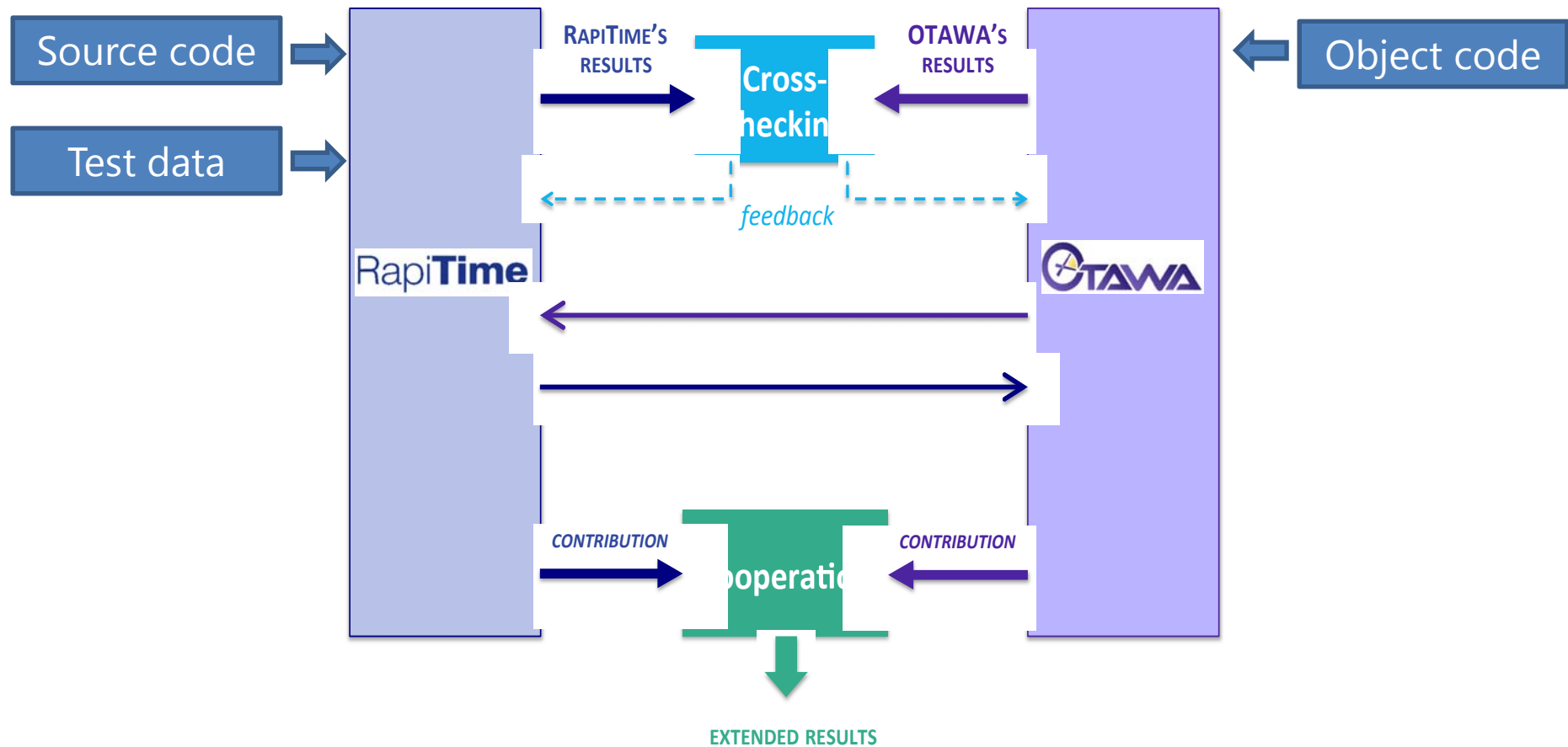
Waiting time calculations involving:

- Critical sections, barrier synchronisation etc

WCET: worst case execution times

“High water marks” (maximum measurements), profiling etc.





- Software verification for multicore is challenging
 - We focus on the timing and profiling parts
- In parMERASA, having some control over the application and the hardware makes this easier
- General multi-core timing analysis is hard, but achievable

parMERASA

The research leading to these results has received funding from the European Community's Seventh Framework Programme (FP7/2007-2013) under grant agreement n° 287519, parMERASA.