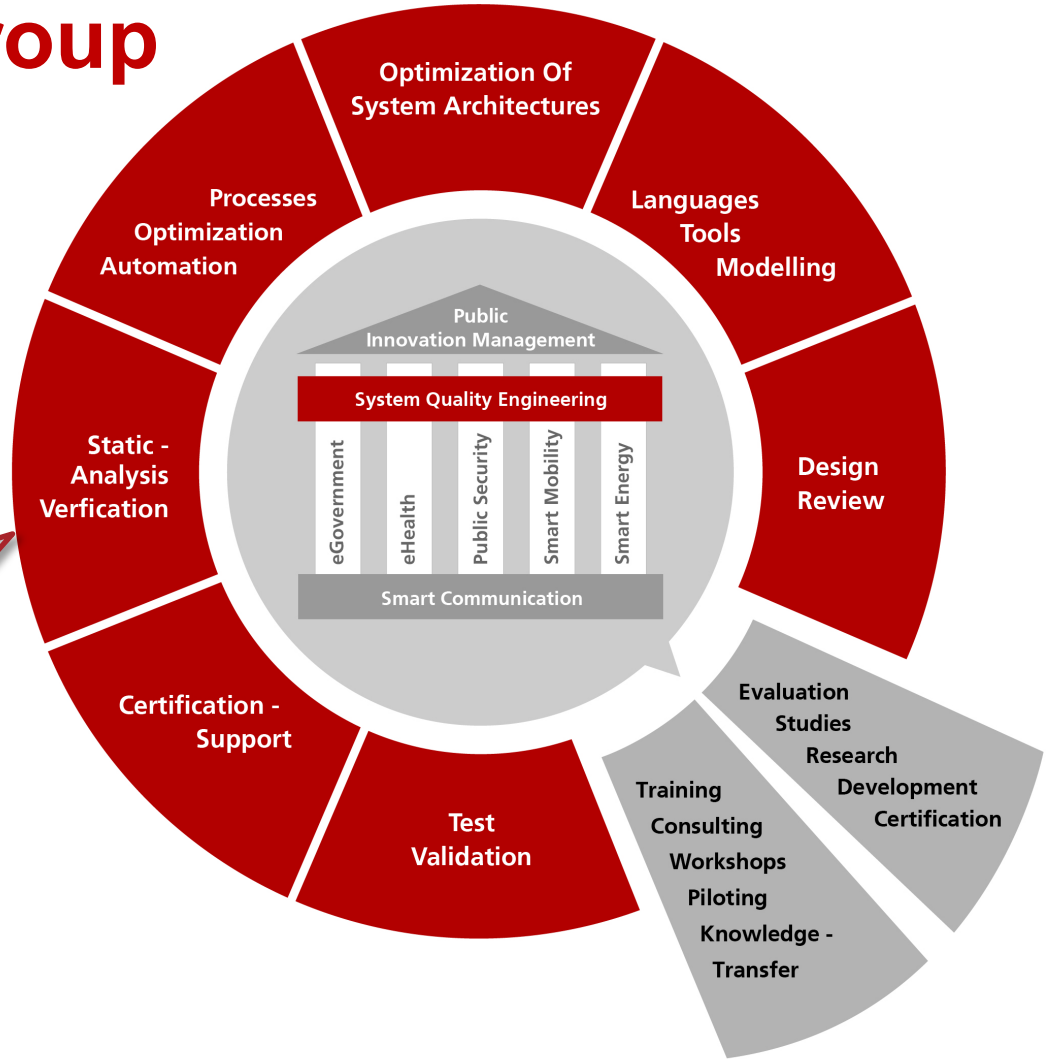
System Quality Center
Verification Group



Formal Verification for Security and Safety

Dr. Jens Gerlach

Problem Space

For some applications we need guarantees that software satisfies its specification in ALL cases.

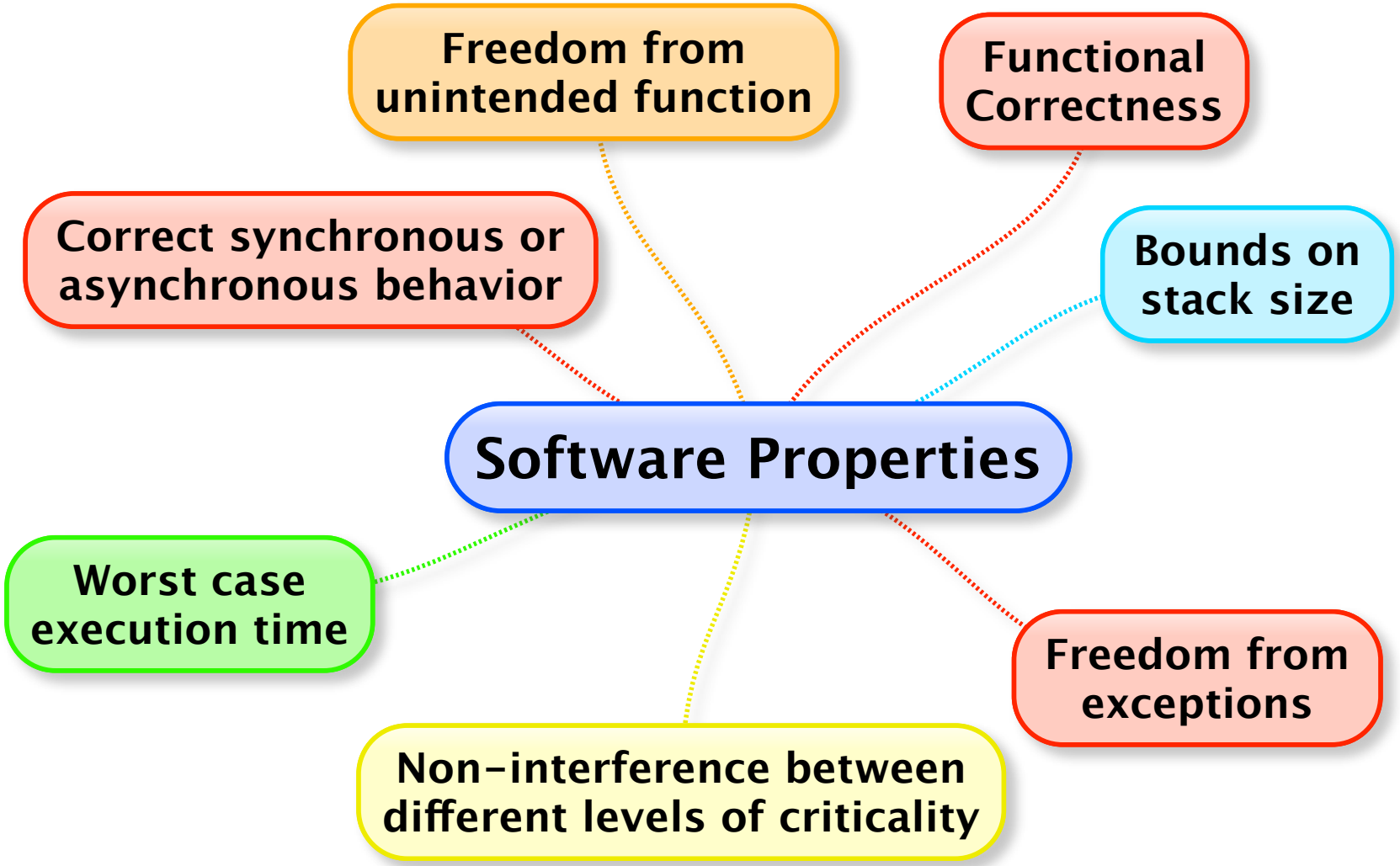
Analysis and Verification of Bitwalker

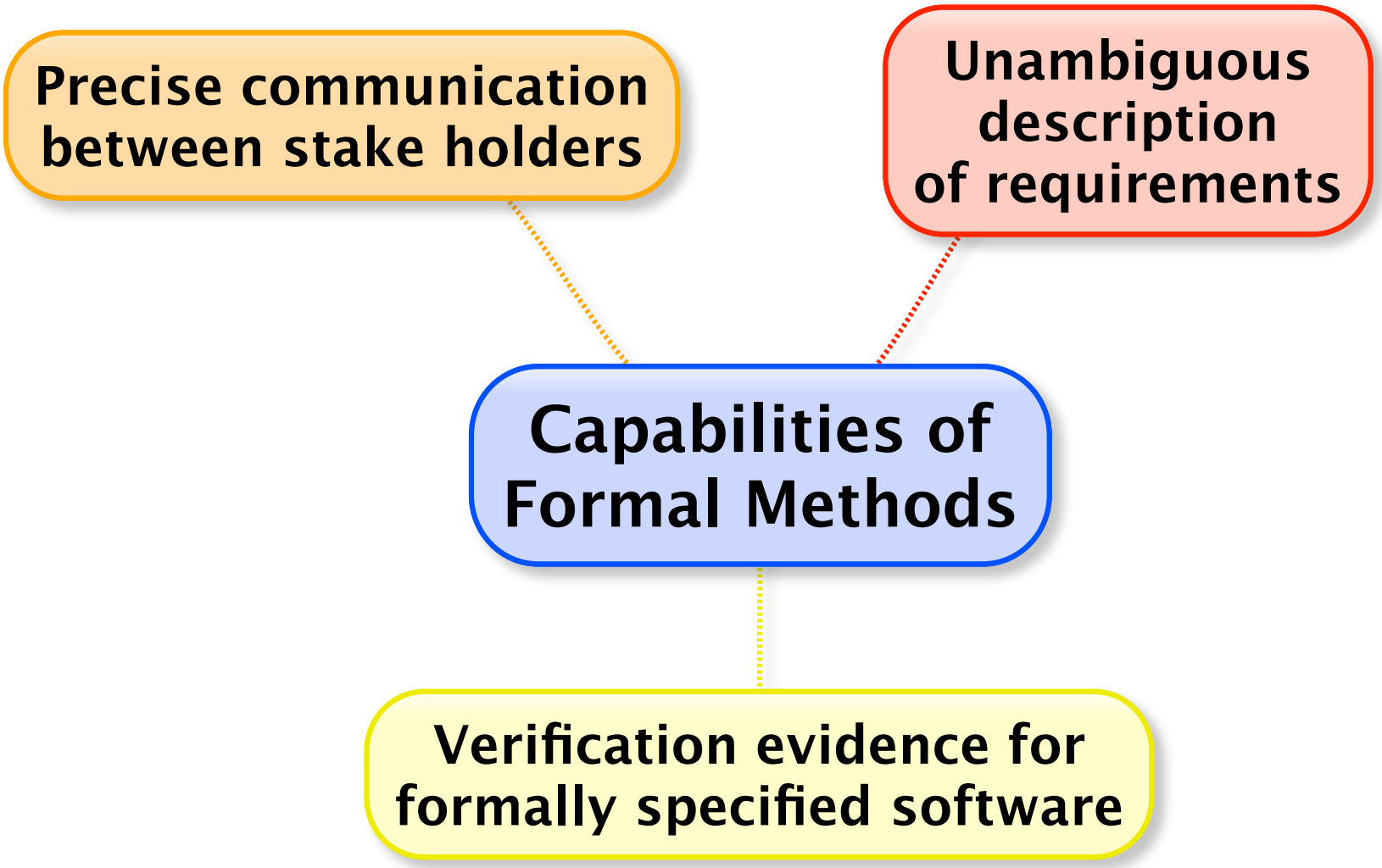
- Bitwalker is a small component written in C
  - extraction and insertion of 64-bit integers from a larger bit-stream

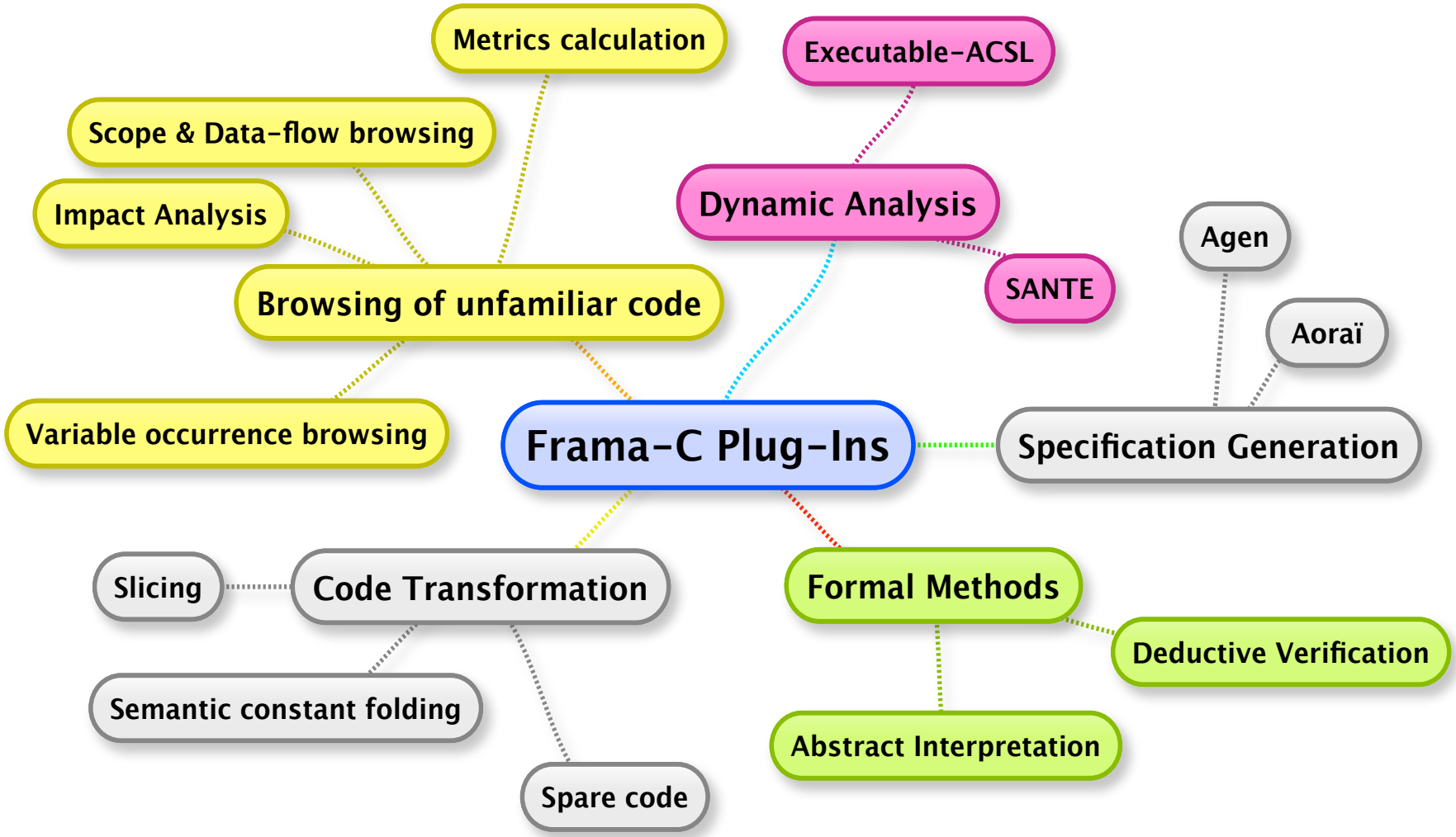
- Partners
  - written by Siemens
  - to be verified by Fraunhofer FOKUS
  - tools provided by CEA LIST

Bitwalker is also an Exercise in …

- eliciting information
- working with incomplete specifications
- investigating the relationship between formal proof and testing with EN 50128
- generating formal C-specifications from the data dictionary (Fraunhofer FOKUS and Fraunhofer ESK)


Freedom from unintended function
Functional Correctness
Correct synchronous or asynchronous behavior
Bounds on stack size
Software Properties
Worst case execution time
Freedom from exceptions
Non-interference between different levels of criticality


Precise communication between stake holders
Unambiguous description of requirements
Capabilities of Formal Methods
Verification evidence for formally specified software


Metrics calculation
Executable-ACSL
Scope & Data-flow browsing
Impact Analysis
Dynamic Analysis
Agen
SANTE
Browsing of unfamiliar code
Aoraï
Variable occurrence browsing
Frama-C Plug-Ins
Specification Generation
Slicing
Code Transformation
Formal Methods
Deductive Verification
Semantic constant folding
Abstract Interpretation
Spare code

General Challenges

- Bitwalker allows us to explore to what extend formal code verification can be used for SIL 4 railway software
  - compare with DO-333 from aeronautics domain

- Bitwalker uses a lot of low-level (including platform-dependent) features of C
  - notably: bit-operations, endianess

Table A 5 from EN-50128

TECHNIQUE/MEASURE	Ref	SIL 0	SIL 1	SIL 2	SIL 3	SIL 4
1. Formal Proof	D.29	-	R	R	HR	HR
2. Static Analysis	Table A.19	-	HR	HR	HR	HR
3. Dynamic Analysis and Testing	Table A.13	-	HR	HR	HR	HR
4. Metrics	D.37	-	R	R	R	R
5. Traceability	D.58	R	HR	HR	M	M
6. Software Error Effect Analysis	D.25	-	R	R	HR	HR
7. Test Coverage for code	Table A.21	R	HR	HR	HR	HR
8. Functional/ Black-box Testing	Table A.14	HR	HR	HR	M	M
9. Performance Testing	Table A.18	-	HR	HR	HR	HR
10. Interface Testing	D.34	HR	HR	HR	HR	HR

Challenges on the Verification Tools

- Frama-C (CEA LIST)
  - allows to apply various verification methods
  - support for bit operations currently not sufficient
  - Frama-C will deal with this in later releases

- Tool qualification
  - might be out of reach for this project