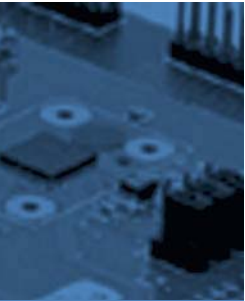


Industrial usage of Formal Methods – an Ericsson view

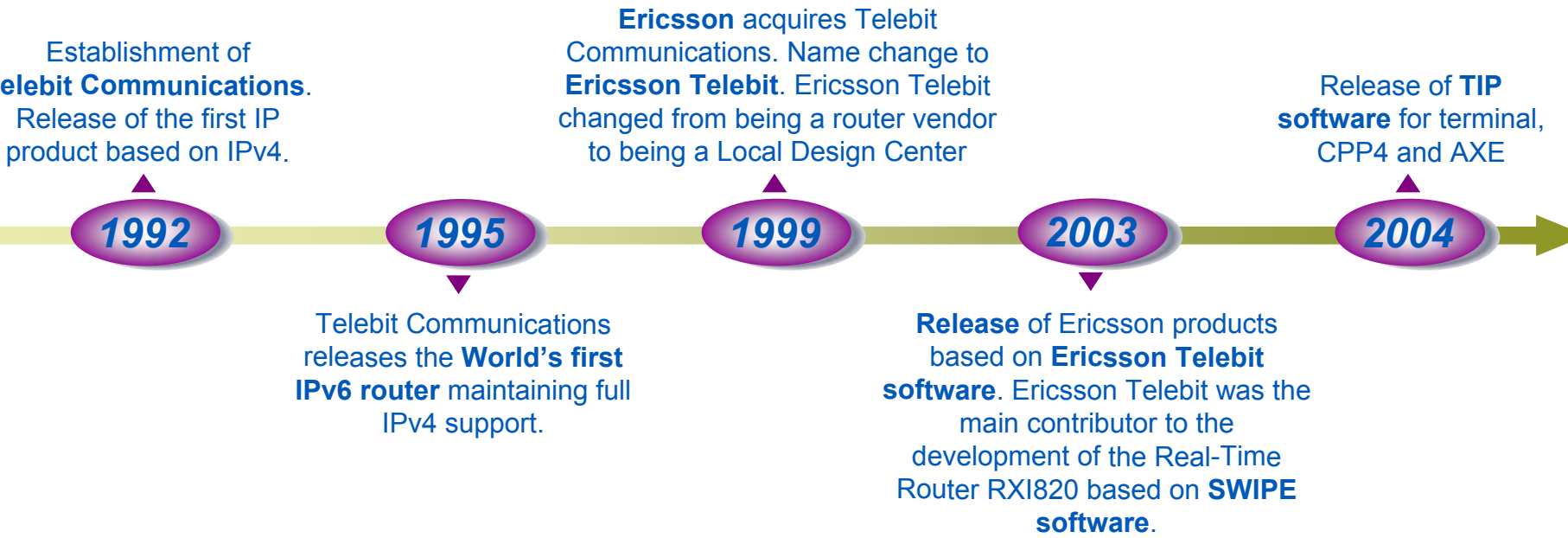
Jens E. Kristensen
jens.kristensen@ericsson.com

About Us




- Ericsson Telebit (TED) is Ericsson's IPv6 company and one of Ericsson's local design centers.
- We deliver IP solutions for 3G terminals and Ericsson's core platforms, AXE, TSP and CPP
- We have the responsibility for TIP (Ericsson's IPv4/IPv6 stack)
- We have the subsystem responsibility for IP Access on CPP
- We participate in the development of Ericsson's Broadband solution EDA (Ericsson Ethernet DSL Access Solution)
- We participate in the development of Ericsson's Integrated Site Solution (Ethernet Switch and SIS Node).
- We participate in extending IPv6 knowledge within and outside Ericsson through our research projects and training activities

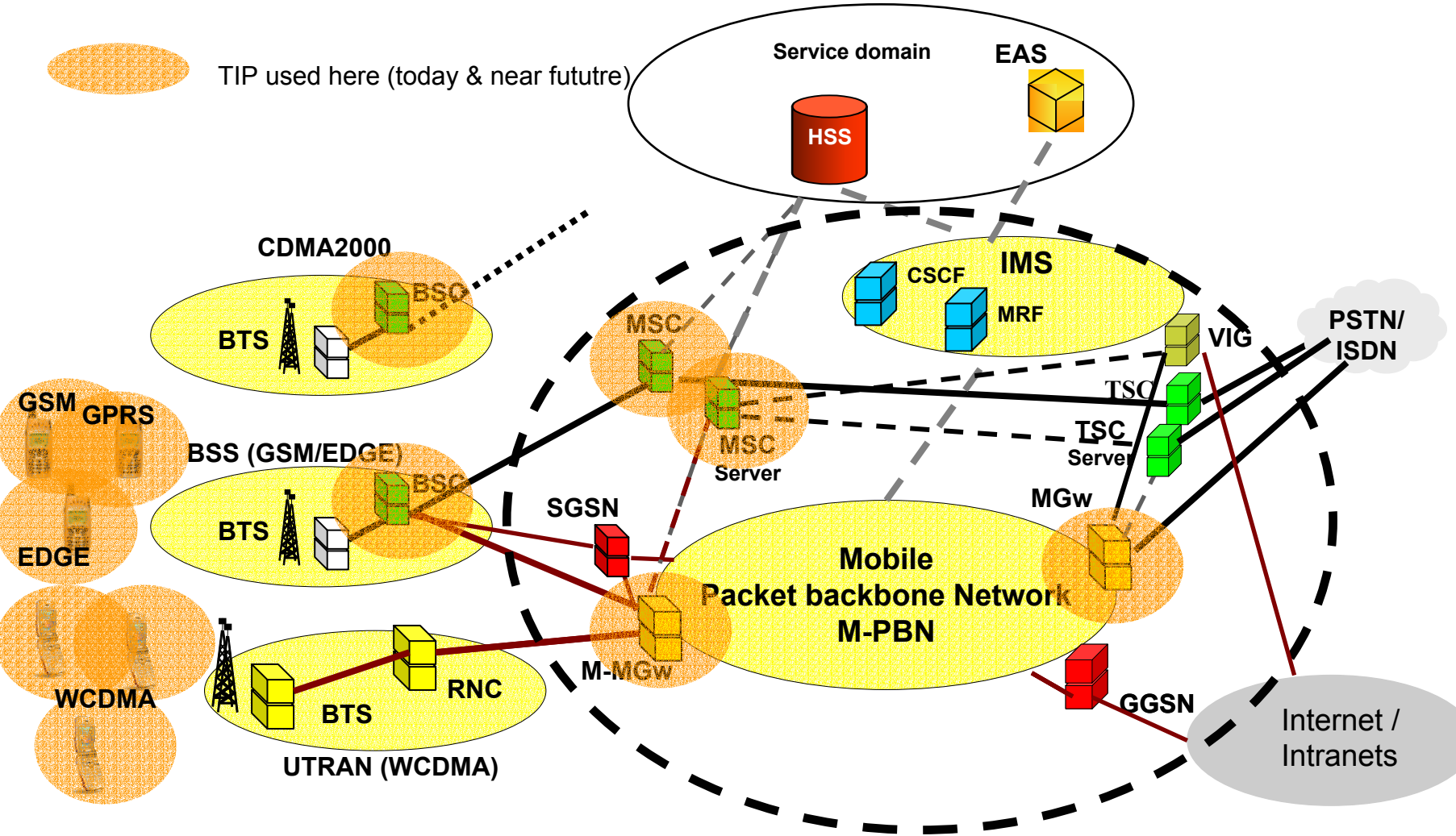
Historical Highlights



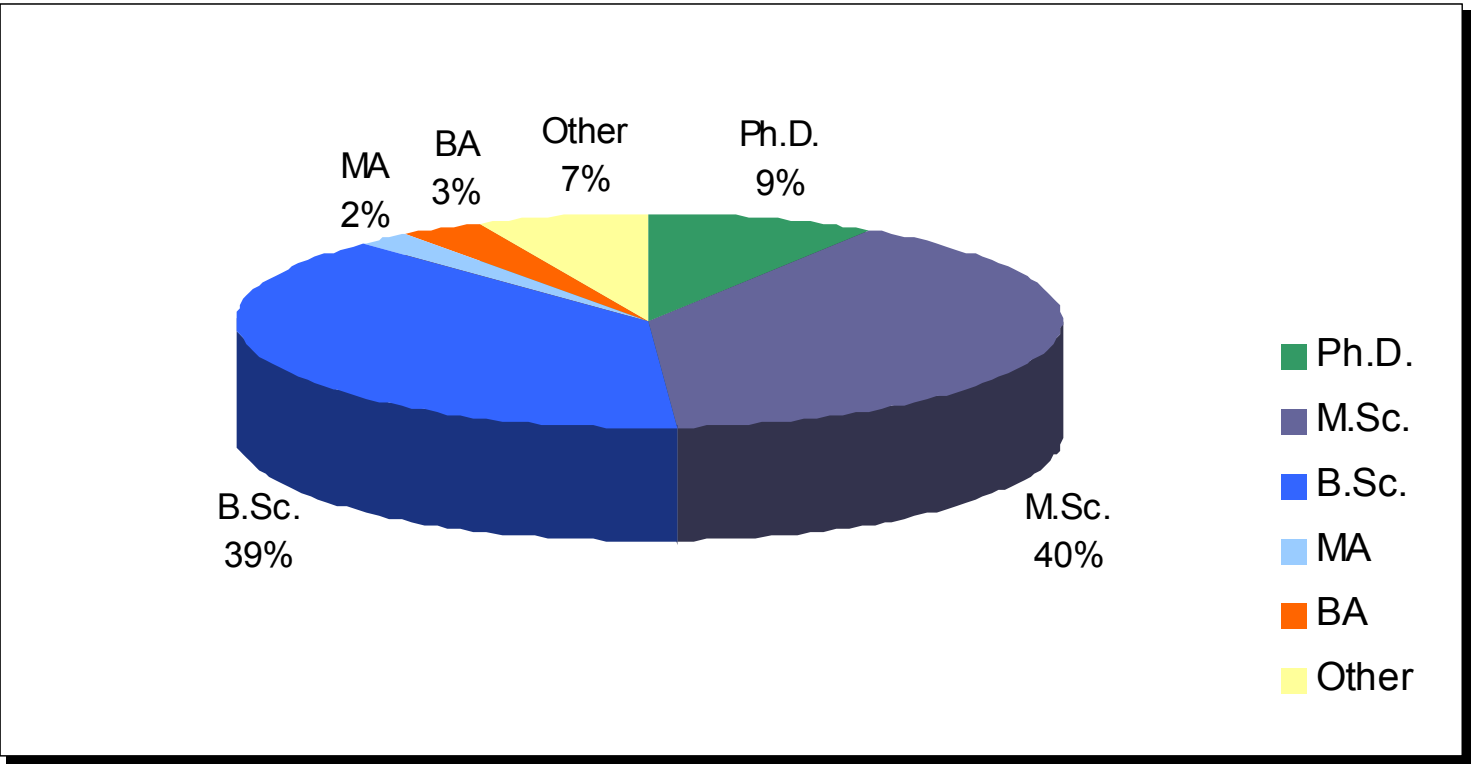


GSM/GPRS/UMTS Architecture

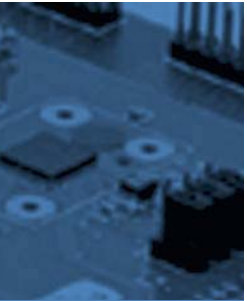
 TIP used here (today & near future)



Educational Level



Quality, Methods & Tools



- ISO9001 certified software development house. The certificate is based on the TED project model adapted from PROPS and SDP-lite and has been chosen as the preferred quality framework.
- Adaptation to Ericsson tools and work methods for configuration management, software modeling, document handling as well as CR and TR handling.
- State-of-the-art integrated software development and test environment based on Linux (this is done on an ad-hoc basis!!!).
- Automated build and test on a daily basis with integrated WEB-based reporting system.
- Application development on CPP based on software development tools (RoseRT).

Quote from an Ericsson Employee

“My experience is that the use of Petri Nets is much limited in software development as a practical use. I'm sure that some flow could be specified that way but never any larger components. By the way, only small, very limited problems, seem to be solvable with formal methods in industry, which tangle with the problems of large complex system. So I doubt its practical use...”

Challenges in software development

- Rate of change, Internet year, Murphy's law
- Complexity is increasing
- Pervasiveness of software in products
- Protocol development is difficult!
 - Concurrency
 - The interaction between the different protocol entities can be very complex
 - Standardisation

Our experience...

- Formal modeling to better communicate with each other
- Executable models would be a great help in understanding the protocols better before we start implementation
- Might be a help in standardisation – conformance testing
- Formal verification is of no help – it is overkill!!!!
- (Formel methods to deal with feature interaction!!!!?????)

Ericsson is in the middle of a transition phase

- Document driven development
 - No standard syntax
 - Different formats (SGML, Word, Framemaker, XML, ASCII)
 - Quality assurance of documents through reviews and inspections
 - Less than 10% of the effort is spent on producing code
- Model driven development
 - Standardized, formal language
 - Analysis and design on a high abstraction level
 - All relevant design information in one place
 - Different views of the same system for different purposes
 - Possibilities for execution of the model
 - Possibilities for automatic code generation

Models promise...

- Overview and details
 - Visual modeling-languages
 - One repository for all design-data
 - Hierachy with “collapse” capabilities
- Architecture focus
 - Bad architecture is more difficult to hide when using visual languages with formal syntax
- Simulation/execution capabilities
 - Proof of concept before all details are decided
- Automatic generation of test cases

Barriers to the introduction of model-driven software development

- Lack of competence – Are formal methods different?
- The integration of methods and tools into the overall development process
- We all know what can be done better, but it is more difficult to find out how it should be done better
- How should we deal with legacy