



# Continuous Integration at WCDMA using Erlang Common Test

By Johan Lundberg at Ericsson Radio Systems

# Contents

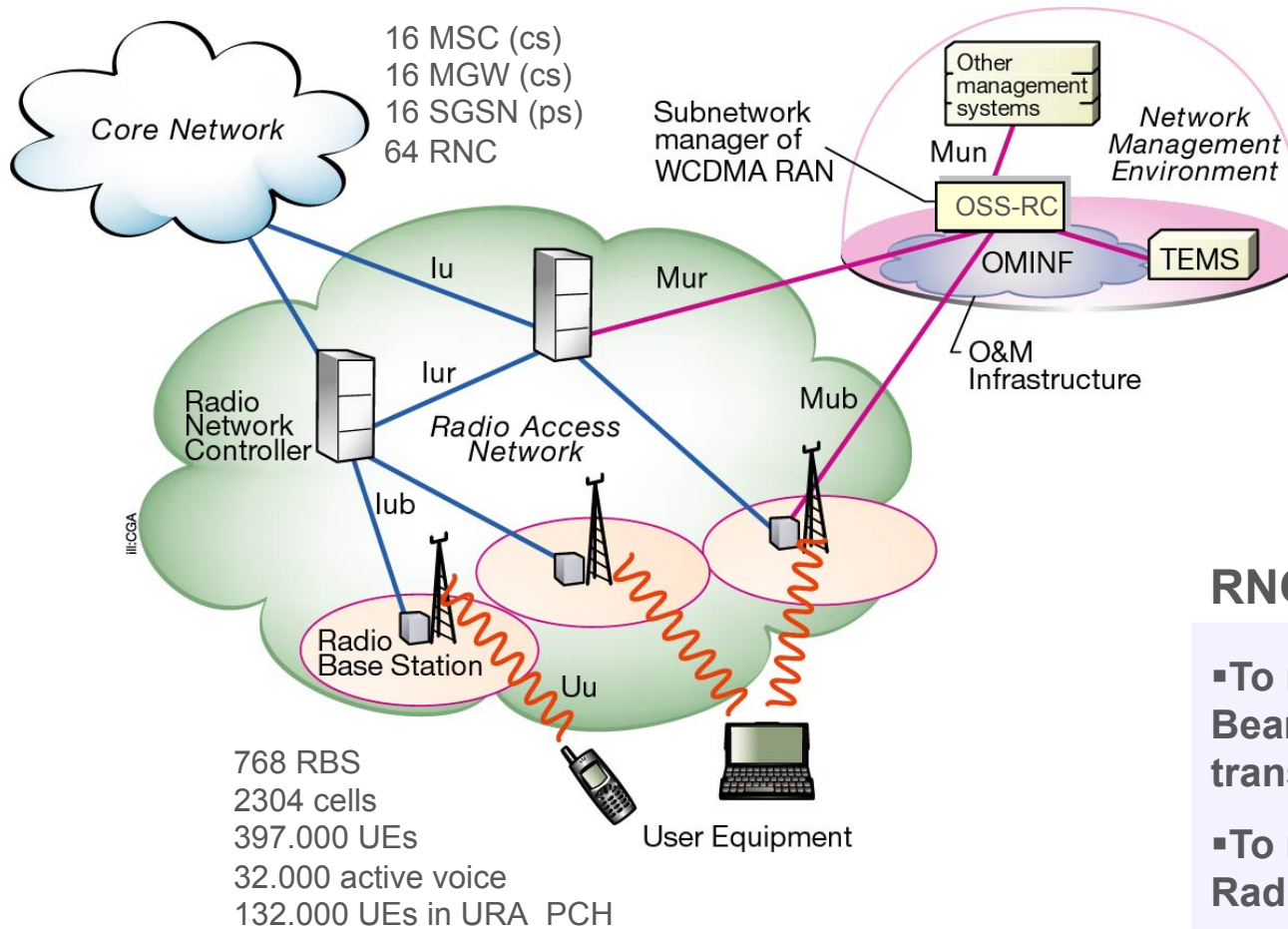
---

- › Introduction- WCDMA overview and transition to CI
- › The ART test framework
- › CI machinery overview
- › Experience from Erlang as test framework

---

# Introduction

# Overview of WCDMA Radio network



## RNC ROLE IN NETWORK

- To manage the Radio Access Bearers for user data transport
- To manage and optimize the Radio Network resources
- To control user mobility

# RNC- brain of the radio network

---

- › Coordinate and manage states in CN, RBS and UE
- › A UE can be in up to 3 cells at the same time
- › A UE can have 100 different Radio Bearer configurations
- › The UE's have many different capabilities and priorities
- › 1000 different performance management counters
- › 100 configurable radio network features
- › Many different HW configurations

# Traditional test phases within RNC

## 1) Unit test

- + Quick feedback, easy to test negative scenarios
- Low level- difficult to map test coverage to full node

## 2) Multi-block (load module) test on target

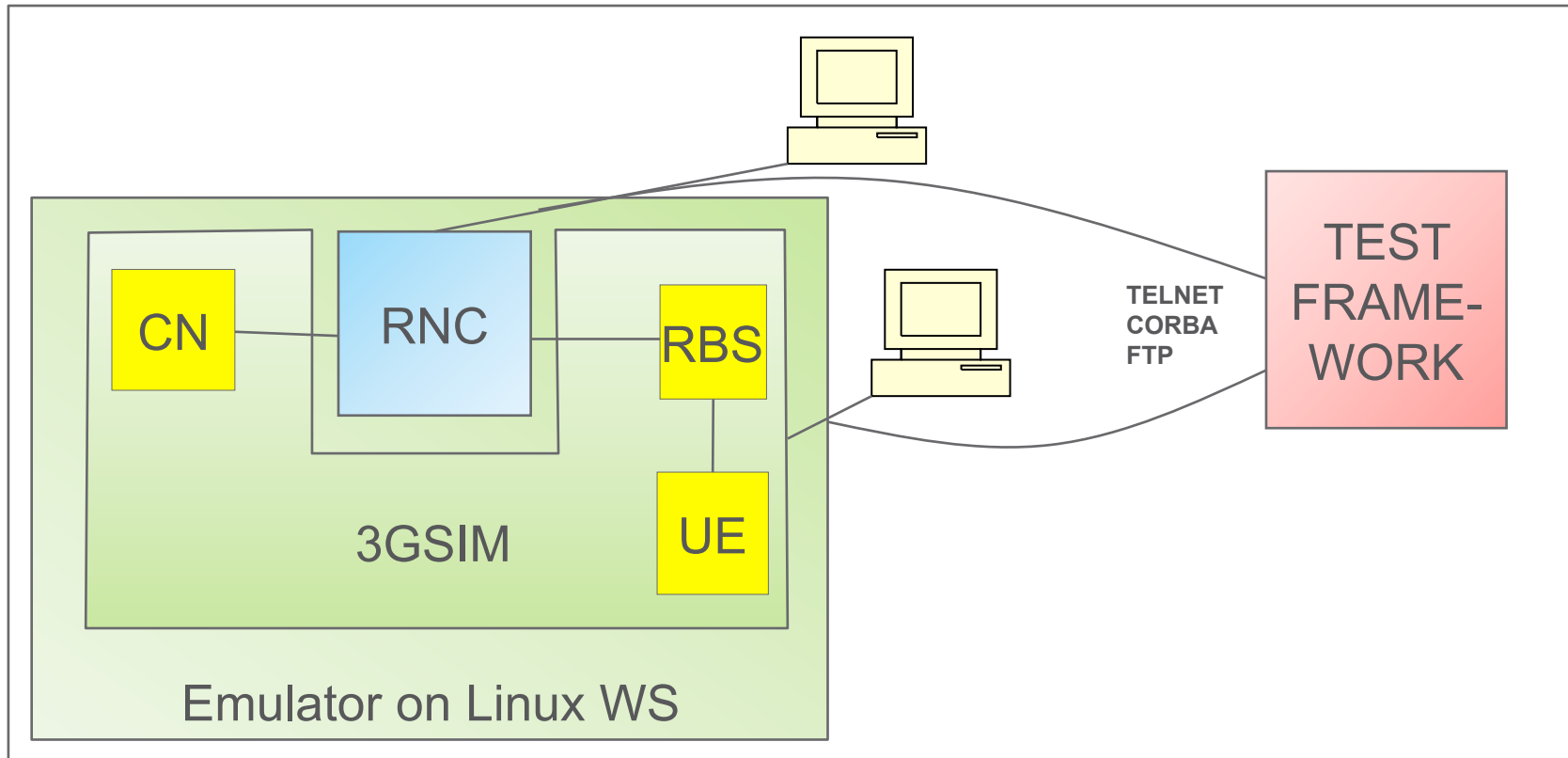
- + Easy to test negative scenarios
  - + Interaction between units is covered
  - Each load module requires a unique test engine
  - All test cases have to be repeated on full node
- + Short lead time to correct faults
  - Low test coverage on node level use cases

## 3) RNC node level test (I&V)

- + Easy to create end user scenarios
  - + Stable simulator using external 3GPP protocols
  - + Both feature test and mass traffic test is possible
  - Less control- difficult to create negative scenarios
- + Mass traffic test uncover many faults
  - Long lead time to correct them
  - Few negative scenarios

# Transition into CI (Continuous Integration)

- How could we get designers to manage node level test environment?
- How could we solve lack of HW?
- How could we make fault isolation easier?

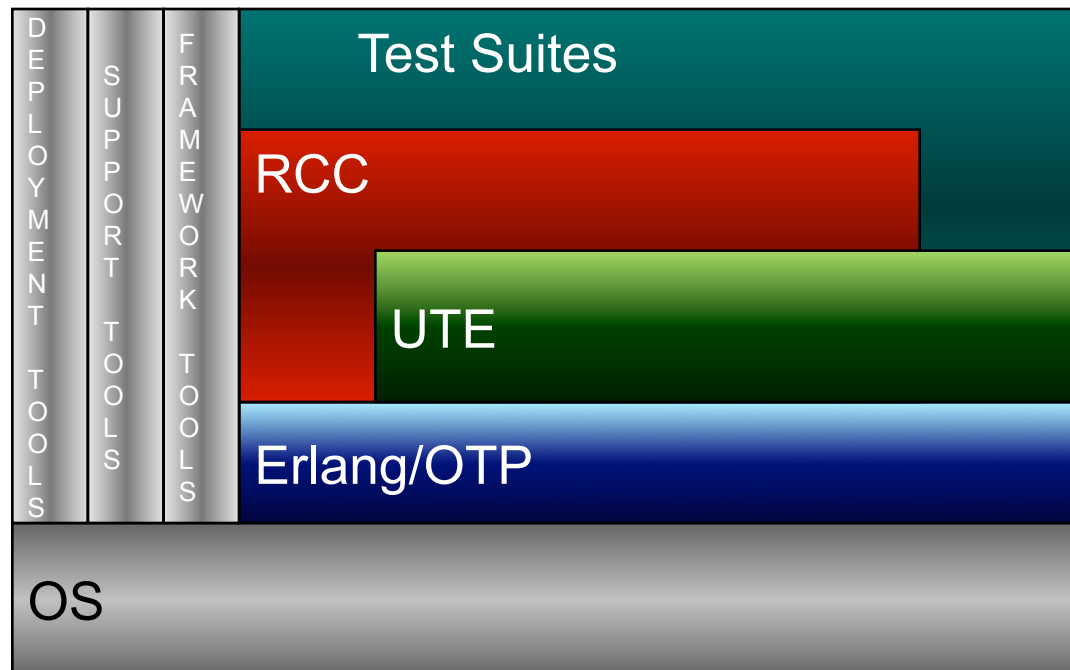


---

# The ART test framework



# ART product overview



# Test case structure and result example

The screenshot displays two main sections: a test suite configuration on the left and its execution log on the right.

### Test Suite Configuration

```

init_per_suite
tc_init
init_per_group
tc soft 1 UeRc 25 EL2
init_per_group
tc soft 1 UeRc 25 EL2
tc soft 1 UeRc 25 EL2
tc soft 1 UeRc 25 EL2
tc soft 1 UeRc 25 EL2
tc soft 1 UeRc 25 EL2
end_per_group
tc init end
end_per_suite
  
```

### Execution Log

Log Line	Time	Result	Comment
[+] EXCLUDE INSIDE Sent: NRAP-DMM-Radi...			Exclude File used is "/vo... All connection capabilities
[+] DIFFERENCE INSIDE Received: UL-DGCH-			Example of failed test case. Let's look at the log
[+] EXC 1556 UL-DGCH-Message			
[+] OK 1558 integrityCheckInfo			
[+] EXC 1560 messageAuthenticatio			
[+] EXC 1561 rrc-MessageSequenceN			
[+] OK 1563 physicalChannelReconfi			
[+] OK 1565 rrc-TransactionIdent			
[+] EXC 1566 count-C-ActivationTi			
[+] EXC 1567 ul-CounterSynchronisat			
[+] EXC 1569 startList			
[+] DIF			
[+] OK 1572 cn-DomainIdentity ps-domain			
[+] OK 1573 DIFFERENT start-Value '00000000 00000000 0110'B			
[+] EXC to: start-Value '00000000 00000000 0101'B			
[+] EXC			
[+] OK			
1576 cn-DomainIdentity cs-domain			
1577 start-Value '00000000 00000000 0010'B			

### Annotations

- Top Left:** Points to the `init_per_suite` and `tc_init` functions, stating: "Sets up the configuration needed by this test suite. All modifications are recorded and a generic cleanup function is called in `tc_init_end`".
- Top Center:** Points to the `tc soft 1 UeRc 25 EL2` line, stating: "case. Verify that expected scenario is reached by verifying et".
- Top Right:** Points to the `nl_exclud` variable, stating: "Example of failed test case. Let's look at the log".
- Middle Left:** Points to the `EXCLUDE INSIDE` log entry, stating: "- All connection capabilities".
- Middle Right:** Points to the `DIFFERENCE INSIDE` log entry, stating: "Postcheck test cases of types: 1) Compare printout before and after TC, example ADM, RES; 2) Compare with saved baseline logfiles, example ASN, SPBAG and PM; 3) Check for error traces, example userdata (SP errors), exc\_err".

# Key points

---

- › Used both for automated functional tests and stability tests with KPI checks
- › Can be run on both real HW and Linux WS
- › Easy to rerun failed functional tests
- › High level test cases- quick to write test suites and low maintenance
- › Functional tests are very similar to the manual tests performed by testers- they understand the test suites
- › Keep the layered structure!

# Future plans- highlights

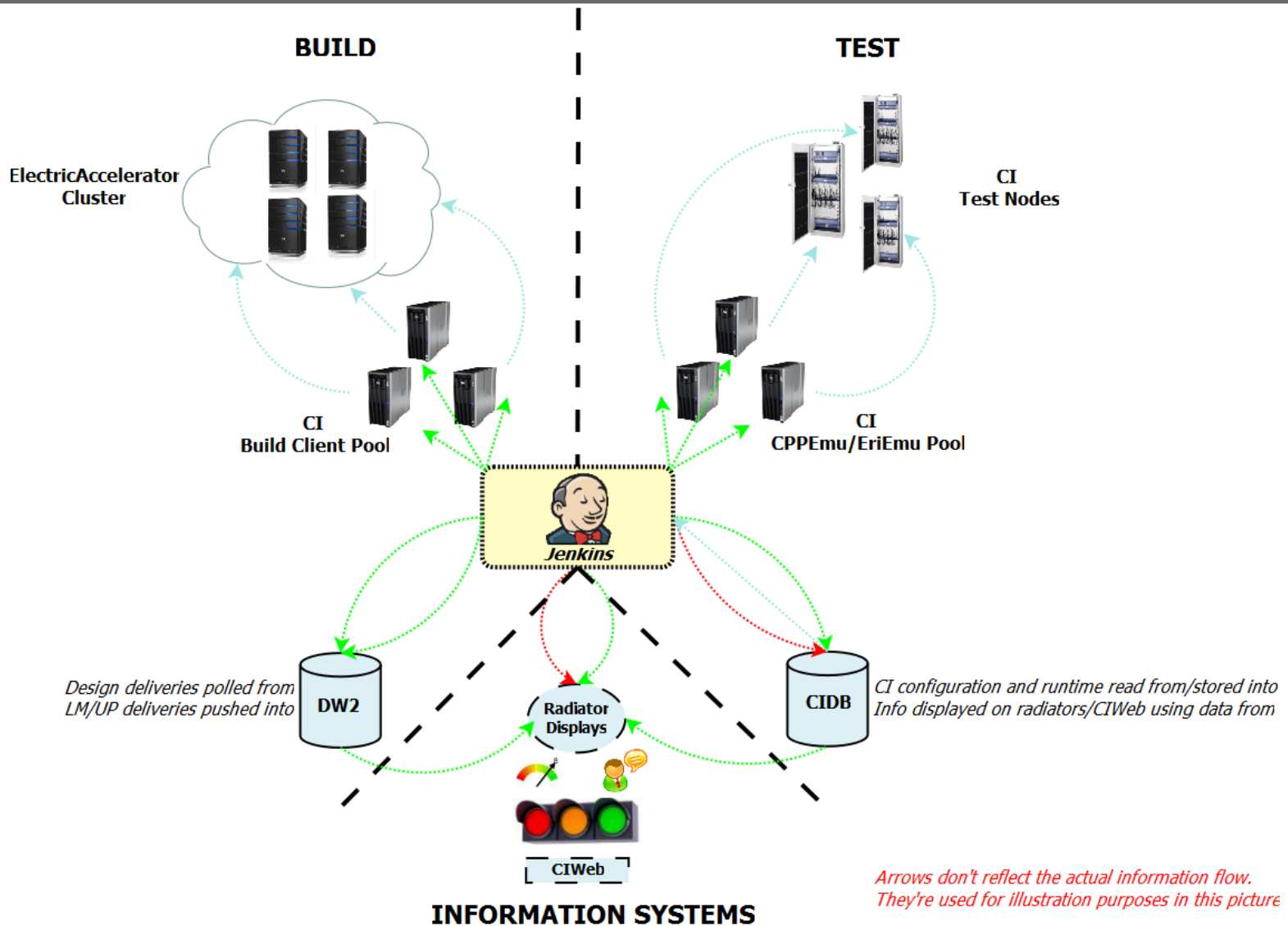
---

- › More negative test scenarios by forwarding protocol messages from 3Gsim to ART and manipulate the response
- › Do peek/ grab of RNC internal protocol messages and manipulate the response
- › Create a test model for the PM counters and validate automatically
- › Requirement on CT to implement support for 8 different log levels. Collect more data and avoid rerunning TCs to trouble shoot
- › Store common test results in an RRS database to manage the amount of test results. HTML logs are not enough.

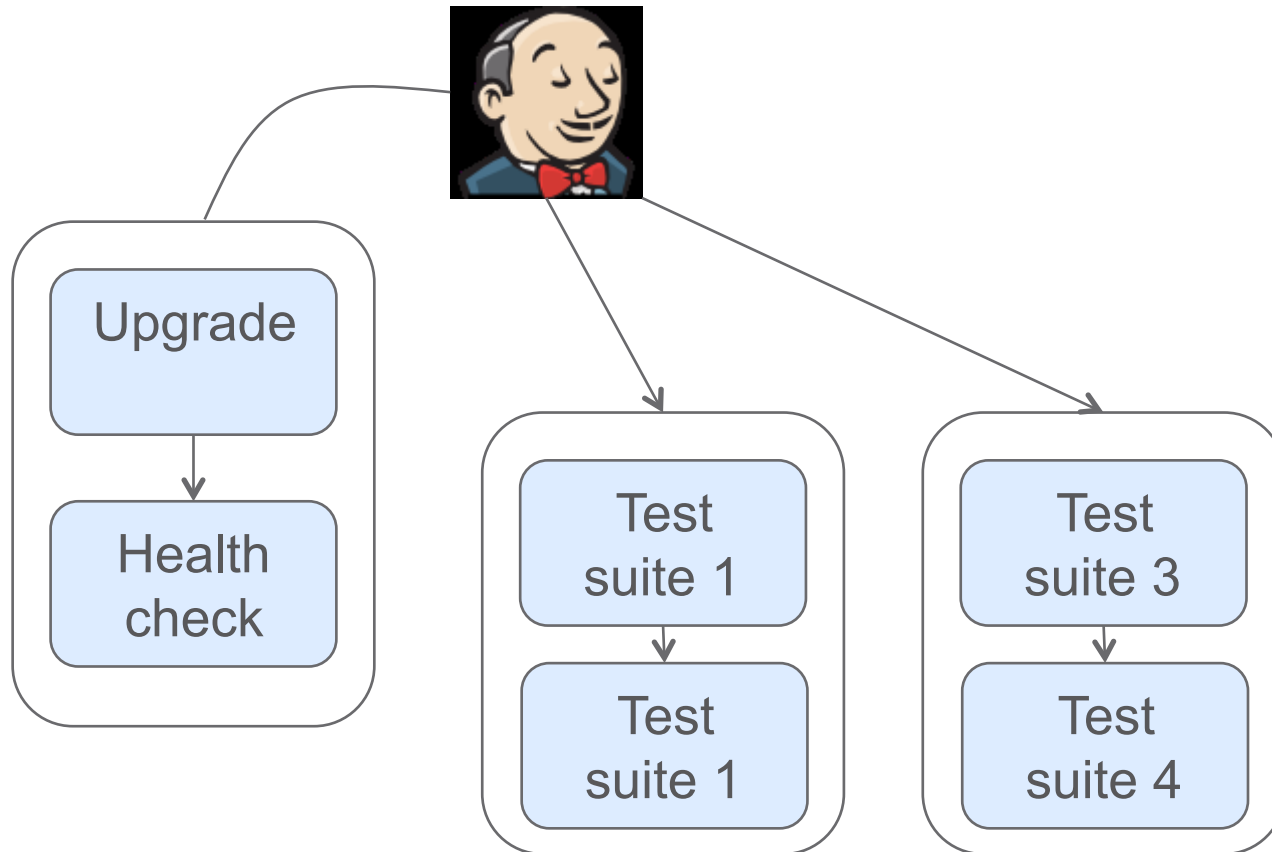
---

# CI machinery overview

# CI machinery overview



# CI machinery overview



Downstream  
projects

---

# Experience from Erlang as test framework



# Experience from Erlang as test framework

---

- › CI has big impact on the IT infrastructure regardless of framework!
- › Erlang is compact, flexible and easy to trouble shoot
- › Pattern matching is really powerful- we can check every action without effort
- › Ability to create patches are really useful
- › Erlang has a reputation of being difficult to learn. But test suites are written on high level
- › We parse a lot of text- not always easy. For example, indexed lists would be nice.
- › It will be difficult to compete with the strong development community for JCAT, which is JAVA based

# Summary

---

- › Introduction- overview of WCDMA and transition to CI
- › The ART test framework
- › CI machinery overview
- › Experience from Erlang as test framework

Email: [johan.lundberg@ericsson.com](mailto:johan.lundberg@ericsson.com)



**ERICSSON**