ARM instrumentation of the PHOENIX ATC system for performance evaluation

Dr. Kai Engels, Stefan Ruppert
MyARM GbR
Neue Straße 4, 63571
Gelnhausen-Rot
Germany
web: http://www.myarm.de
eMail: info@myarm.de

Ralf Heidger
DFS Deutsche Flugsicherung
GmbH
Systemhaus, Dep. SH/T
Am DFS Campus 7, 63225
Langen Germany
eMail: ralf.heidger@dfs.de

San Diego, CMG 2007, 6 December 2007

Contents

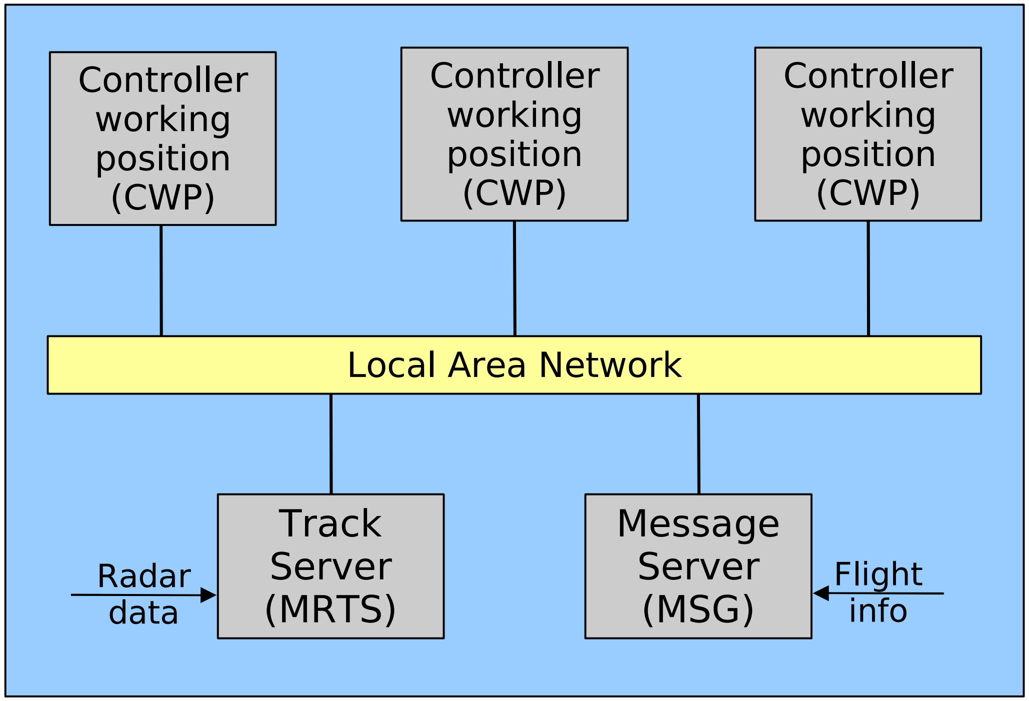
- Introduction
- PHOENIX ATC system
- Test environment and scenarios
- Response time types
- Measurement results
- MyARM environment
- Summary

Introduction

- DFS develops their own ATC systems like PHOENIX
- PHOENIX processes radar data and presents flights to a controller
- ATC systems are subject to performance requirements (1 second requirement)
- ARM measures distributed transactions (Correlators)
- Performance evaluation for two different scenarios (live, artificial)

PHOENIX system architecture

- Client-Server based
- Runs on Linux OS
- Tracker (MRTS) serves tracks by processing incoming radar data
- Message Server (MSG) serves flight information data and links to tracks
- Controller Working Position (CWP) displays the tracks with all information



Controller working position (CWP)

Controller working position (CWP)

Controller working position (CWP)

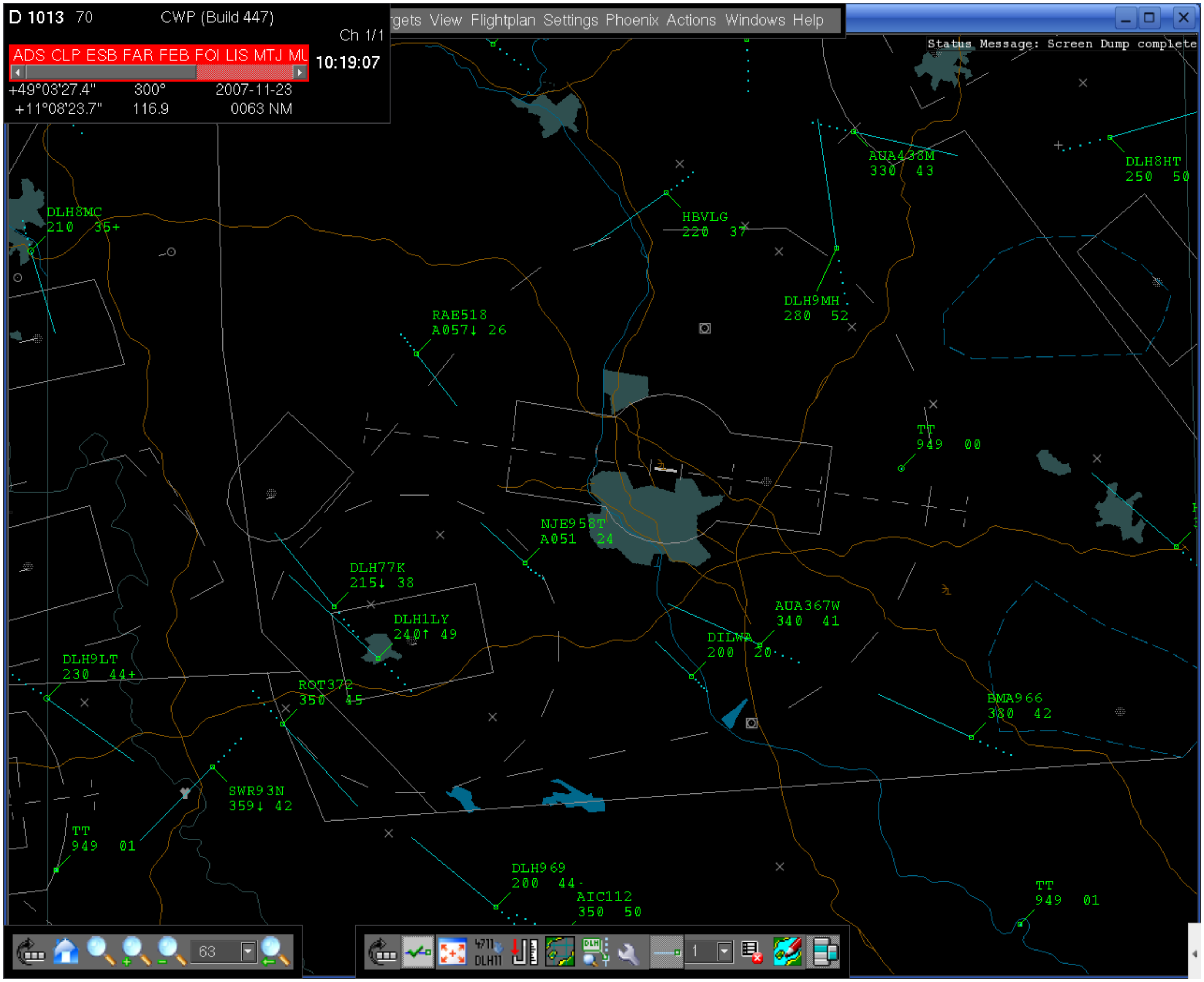
Local Area Network

Radar data

Track Server (MRTS)

Message Server (MSG)

Flight info



D 1013 70

CWP (Build 447)

Ch 1/1

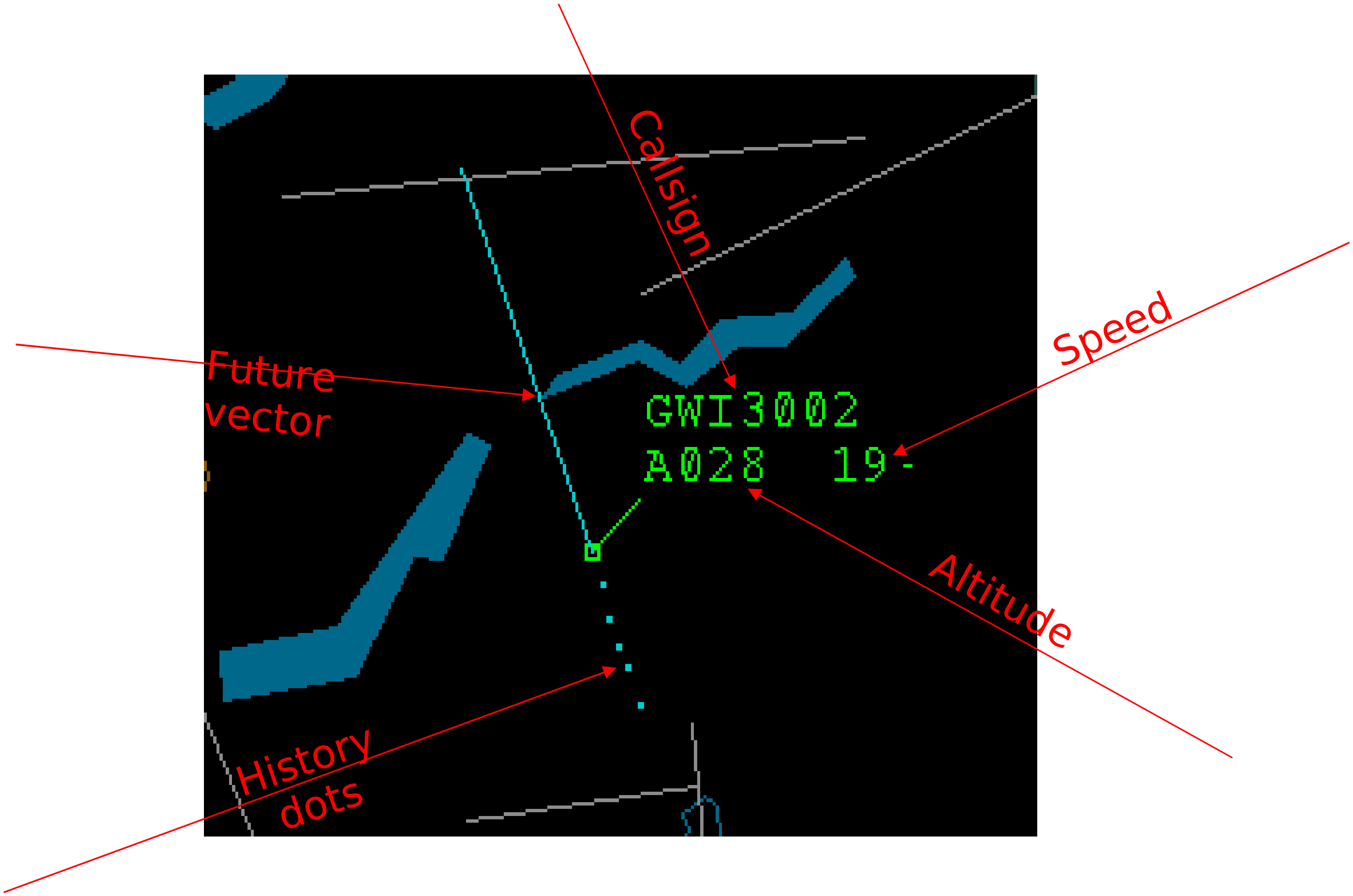
10:19:07

+49°03'27.4" 300° 2007-11-23

+11°08'23.7" 116.9 0063 NM

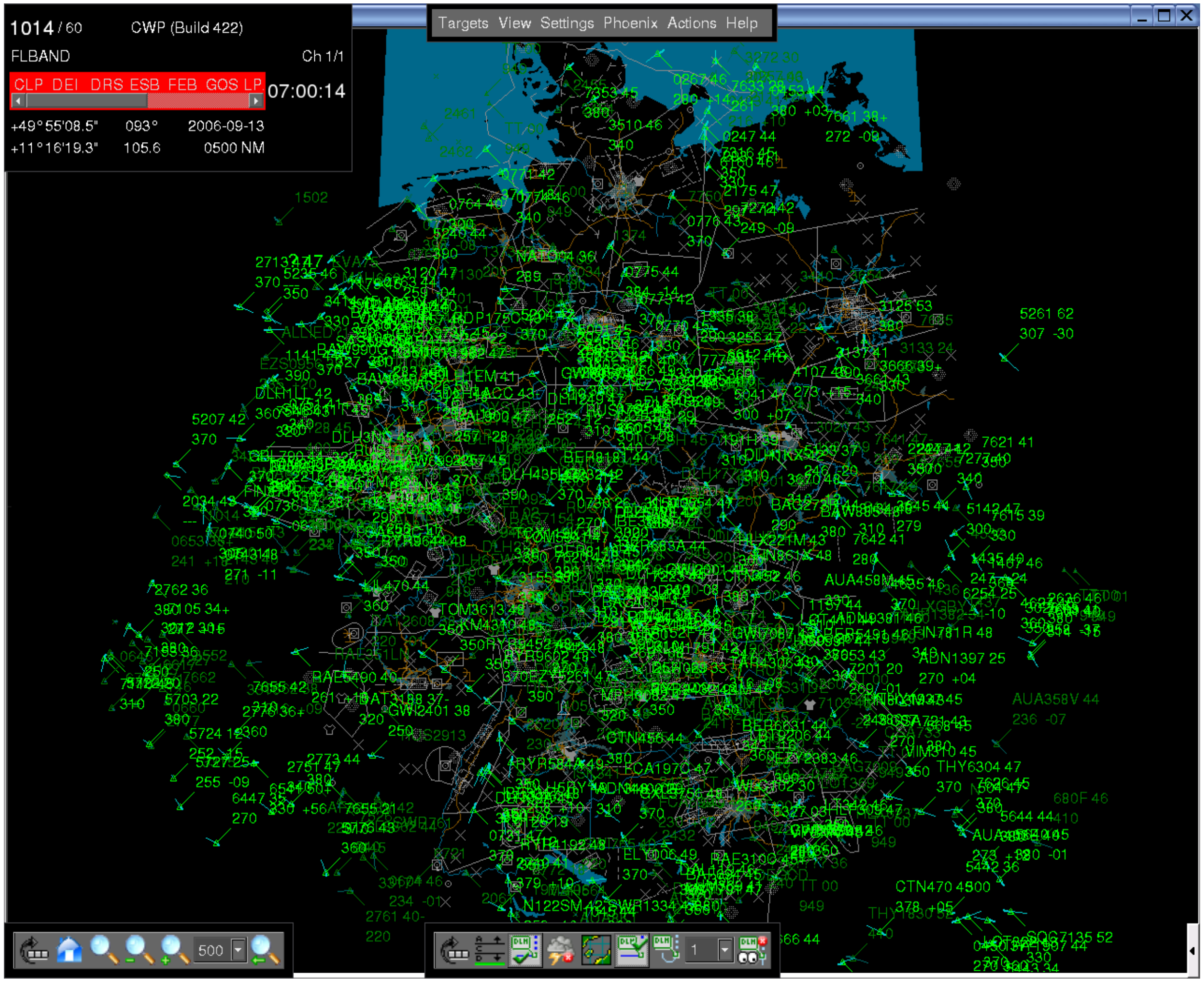
Status Message: Screen Dump complete

PHOENIX track label



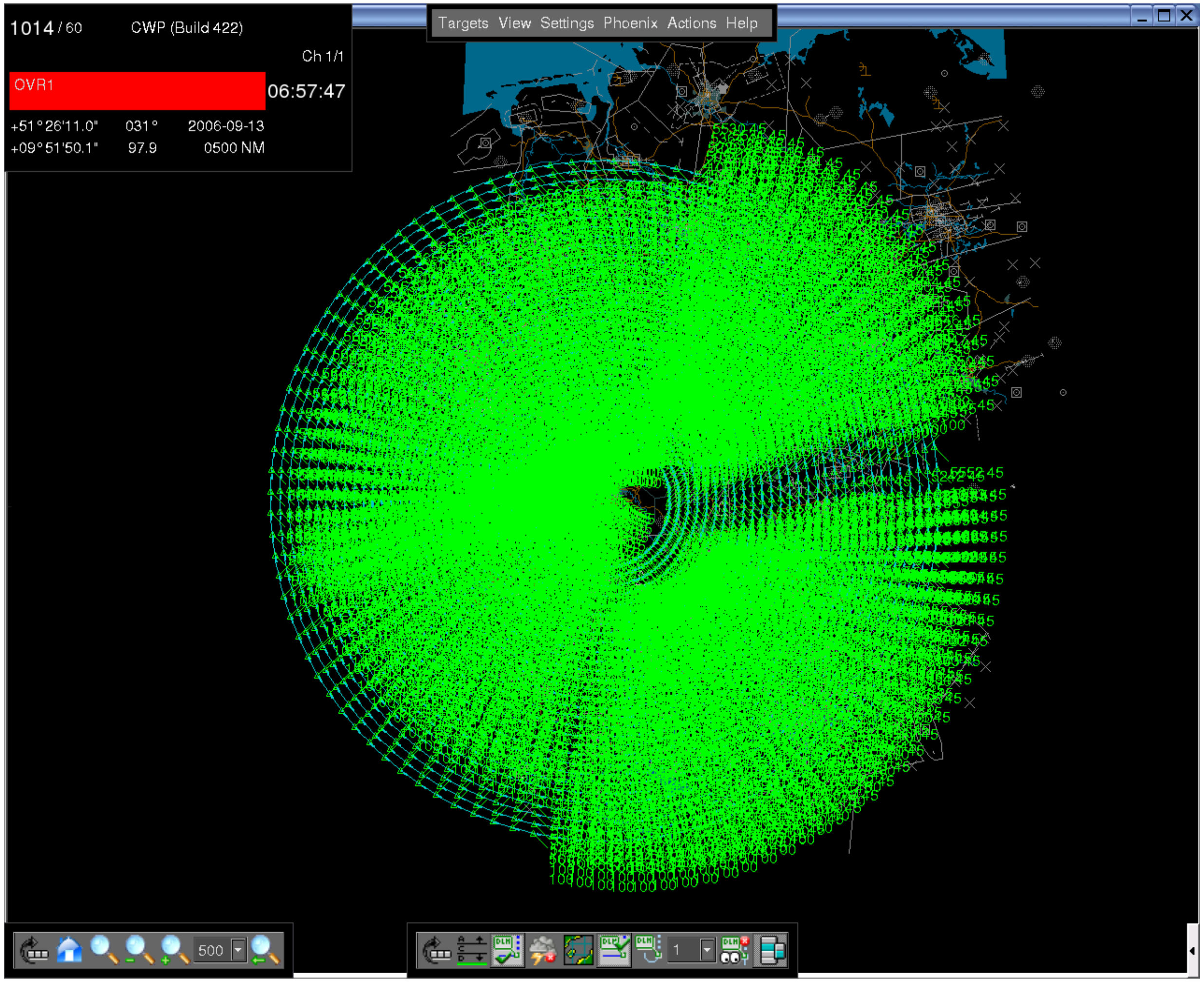
Scenarios (1)

- Live Scenario
  - 30 minutes, 22. June 2006
  - 700 tracks on average



Scenarios (2)

- Artificial scenario, 10 minutes
  - 100 tracks on 30 circles
  - Exactly 3000 tracks on display



1014 / 60 CWP (Build 422)

Ch 1/1

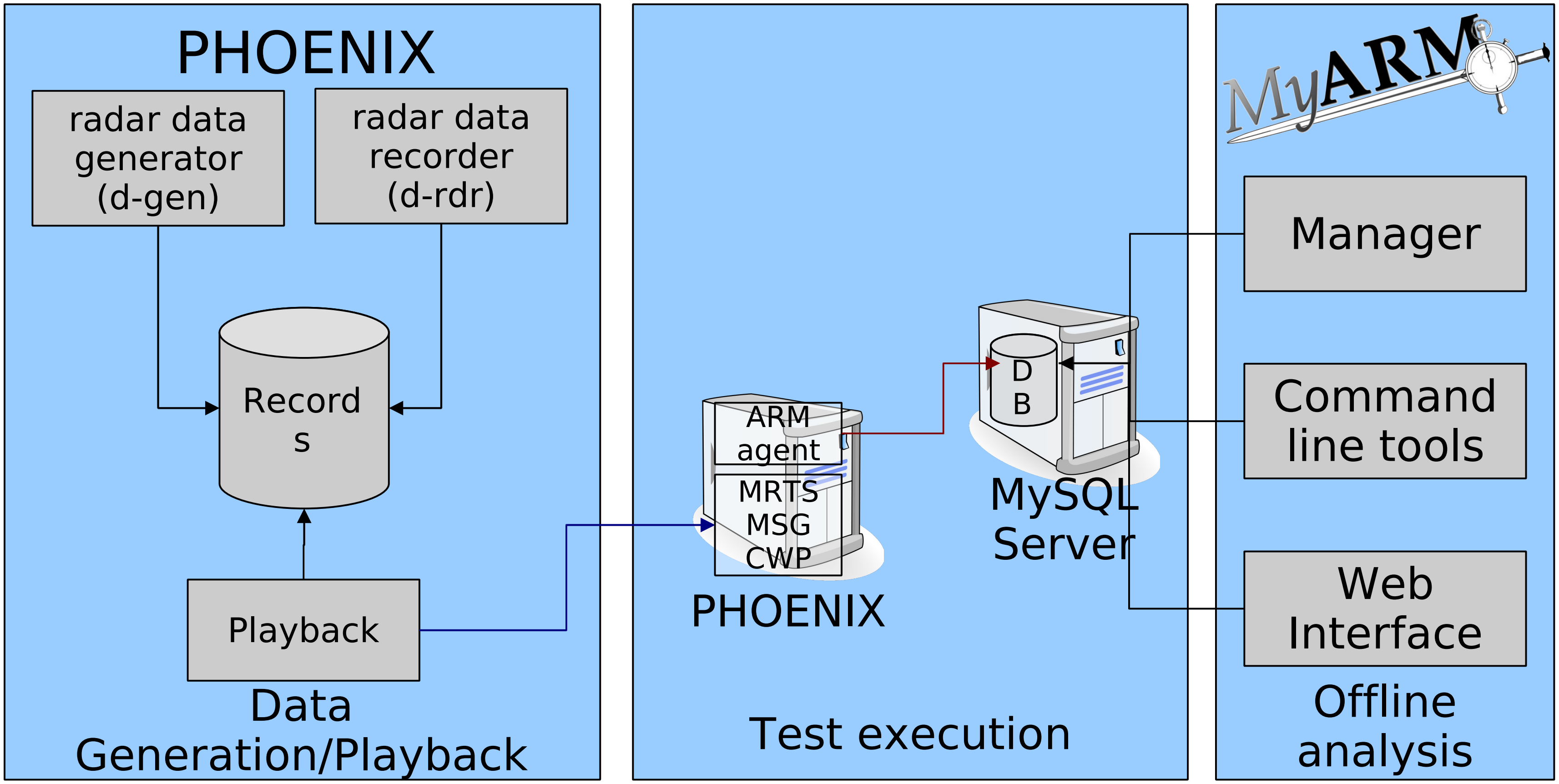
OVR1 06:57:47

+51°26'11.0" 031° 2006-09-13

+09°51'50.1" 97.9 0500 NM

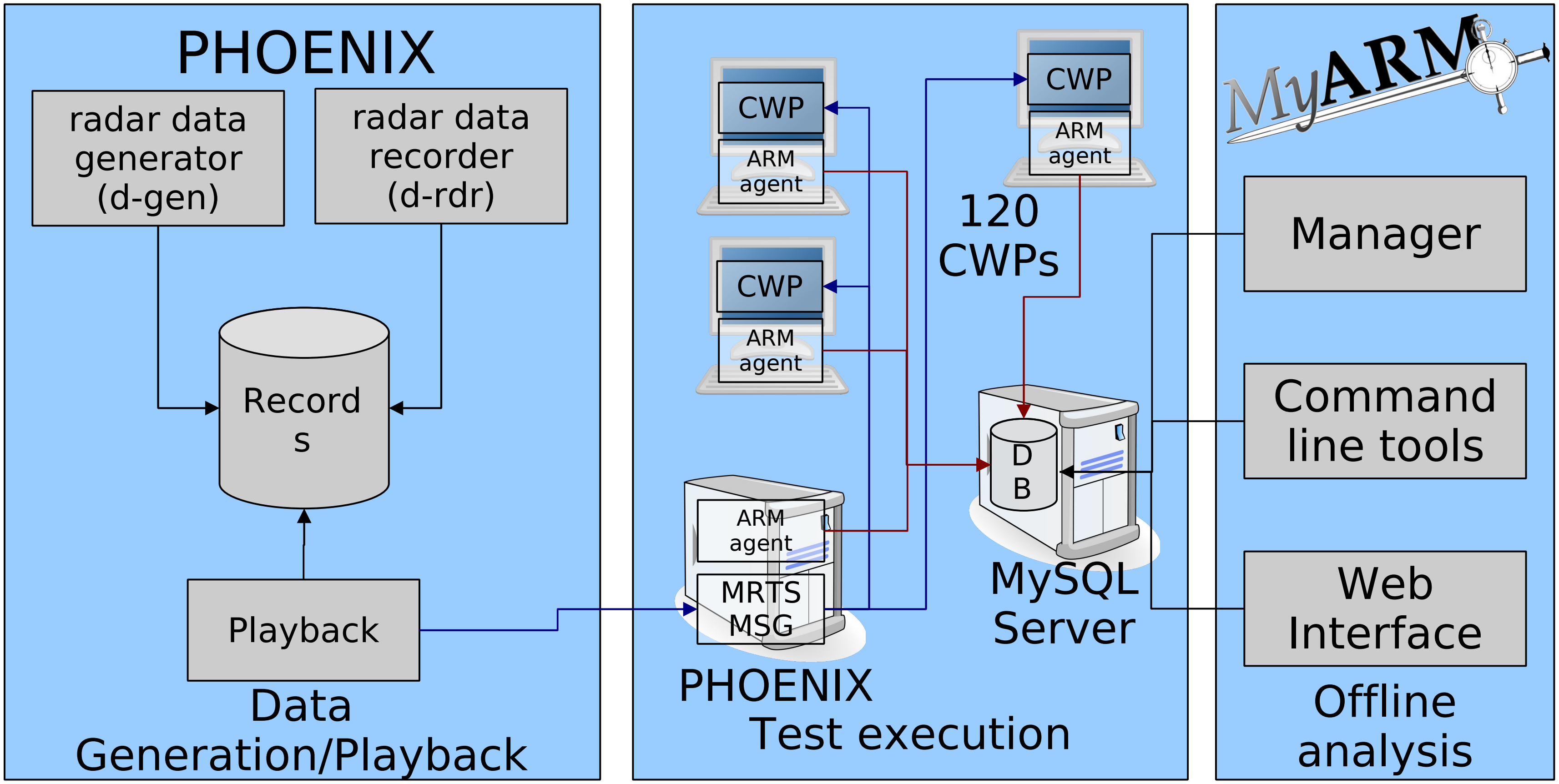
Standalone test environment

- 1 PC with Tracker (MRTS), Message Server (MSG) and CWP



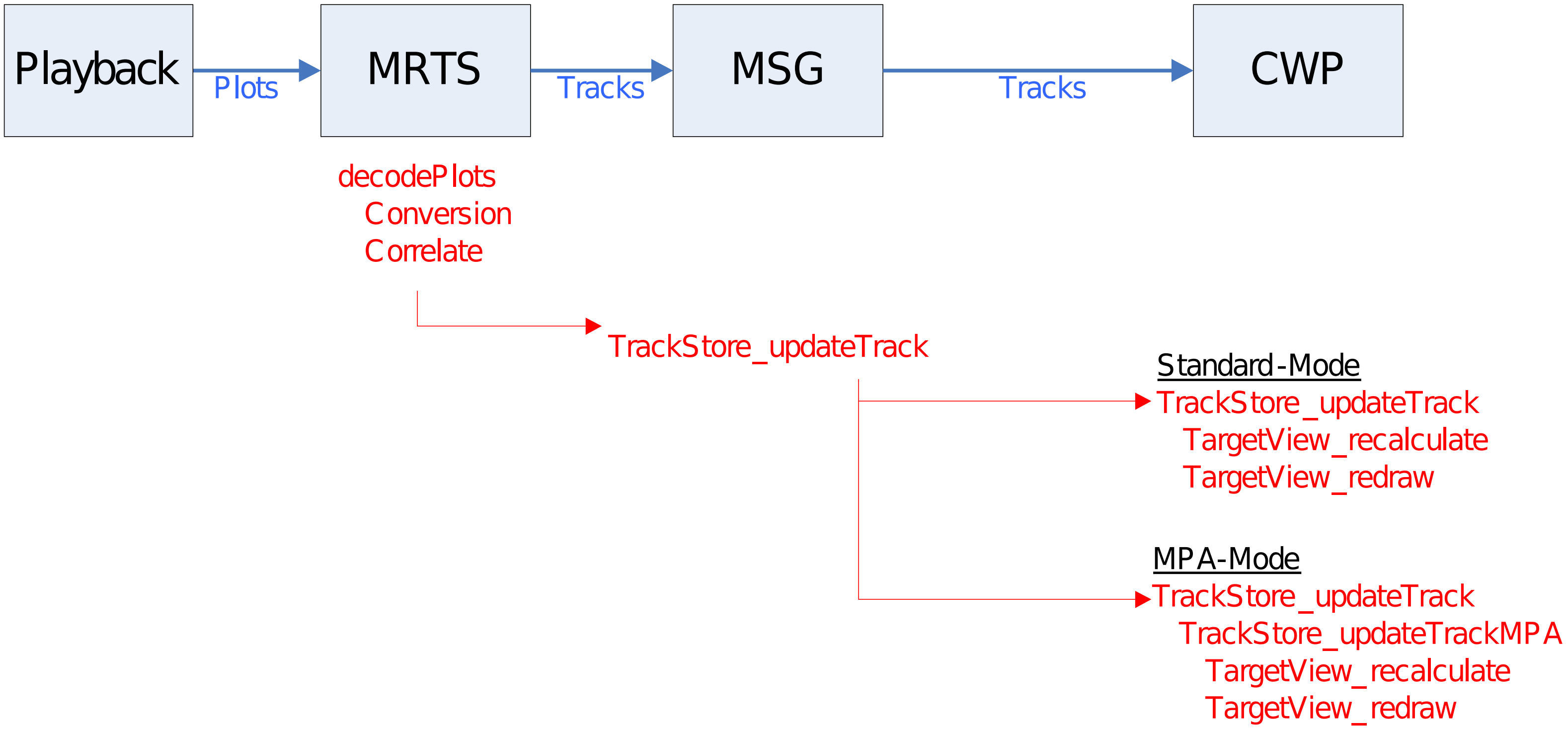
Distributed test environment

- 1 MRTS/MSG Server, 110 CWPs PC, 1 DB Server



PHOENIX ARM instrumentation

- ARM instrumentation points within PHOENIX processes:



Playback

Plots

MRTS

Tracks

MSG

Tracks

CWP

decodePlots
  Conversion
  Correlate

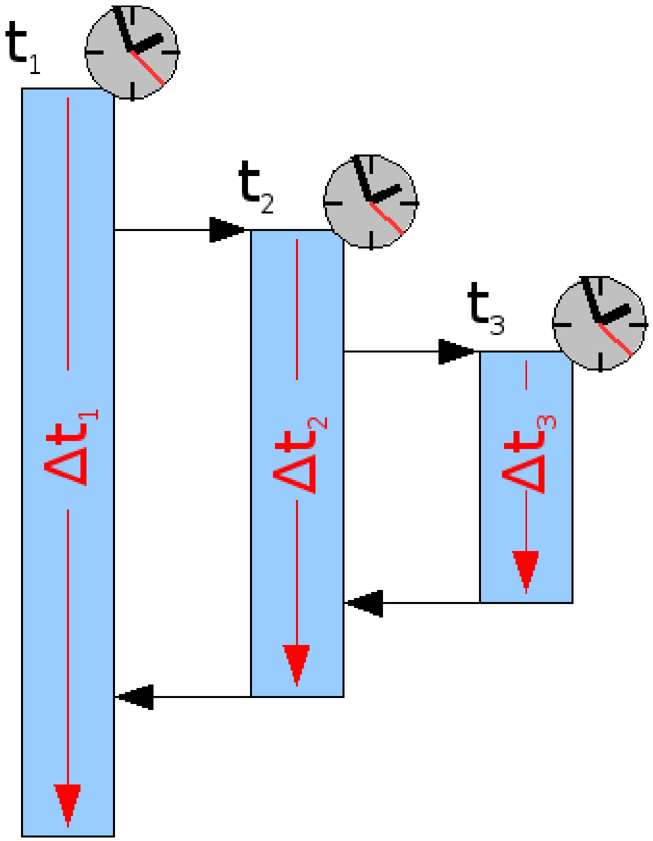
TrackStore_updateTrack

Standard-Mode
TrackStore_updateTrack
  TargetView_recalculate
  TargetView_redraw

MPA-Mode
TrackStore_updateTrack
  TrackStore_updateTrackMPA
    TargetView_recalculate
    TargetView_redraw

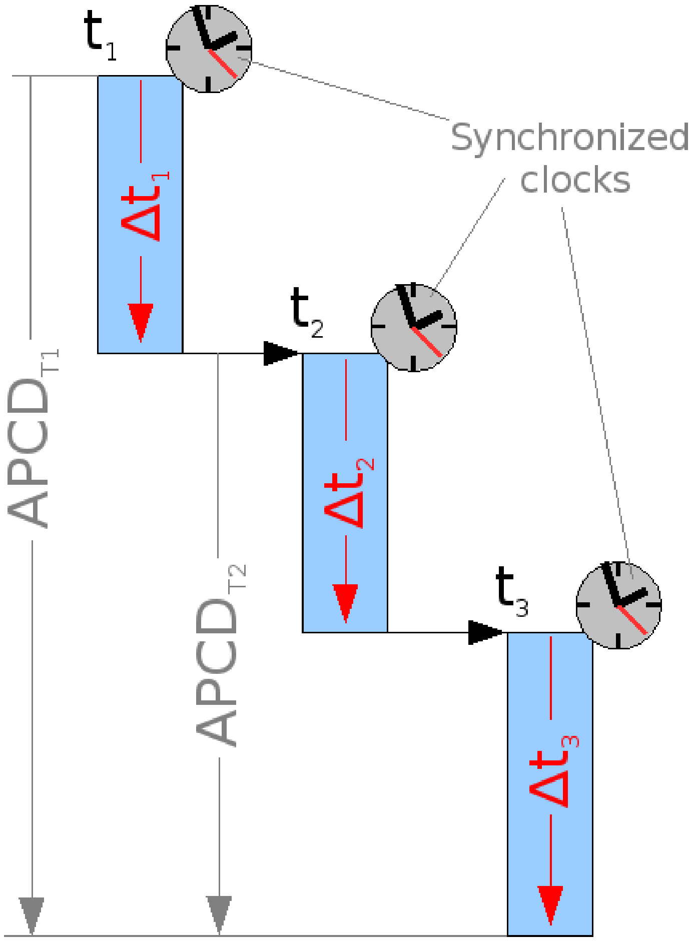
Response time types (overview)

- Synchronous response times (ARM 4.0)
- Asynchronous response times (ARM 4.1)
  - Asynchronous Parent-Child Duration (APCD)
  - Asynchronous Child-Parent Duration (ACPD)



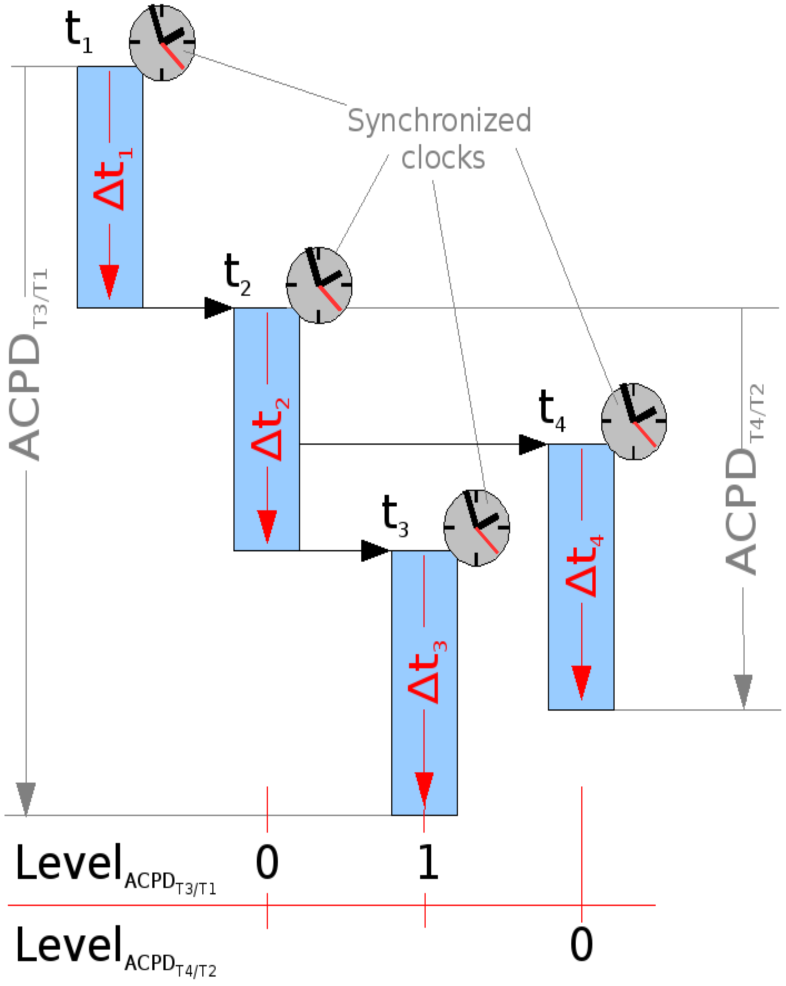
Response time types (APCD)

- Transaction start times
  - t_1 MRTS decodePlots
  - t_2 MSG updateTrack
  - t_3 CWP redrawTarget
- Asynchronous Parent-Child Duration of MRTS-CWP:
  - $APCD_{T1} = t_3 + \Delta t_3 - t_1$
  - only once per



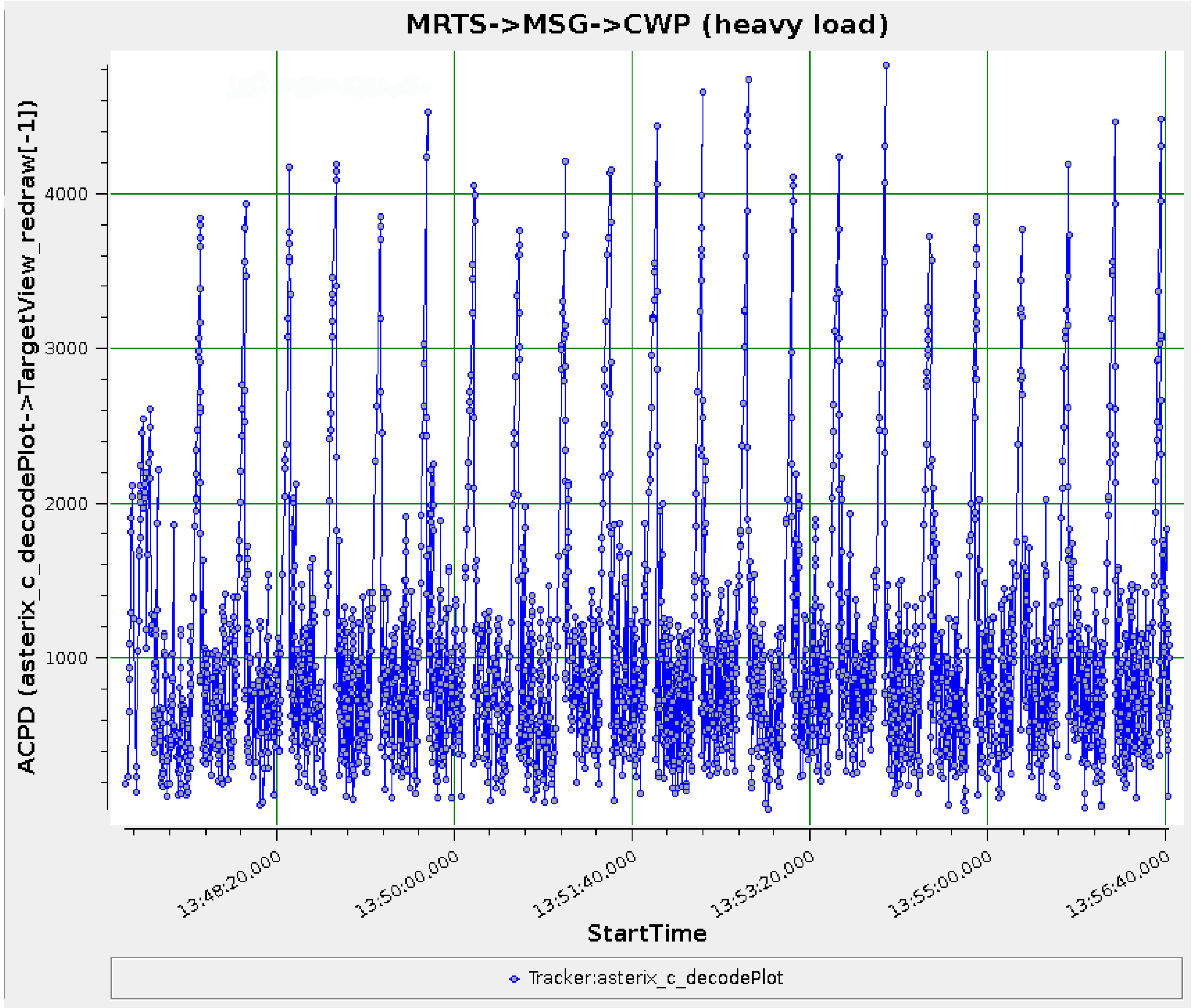
Response time types (ACPD)

- Transaction start times
  - t_1 MRTS decodePlots
  - t_2 MSG updateTrack
  - t_3 CWP redrawTarget
- Asynchronous Child-Parent Duration of CWP-MRTS:
  - $ACPD_{T3/T1} = t_3 + \Delta t_3 - t_1$
  - multiple times for a parent = once for each child



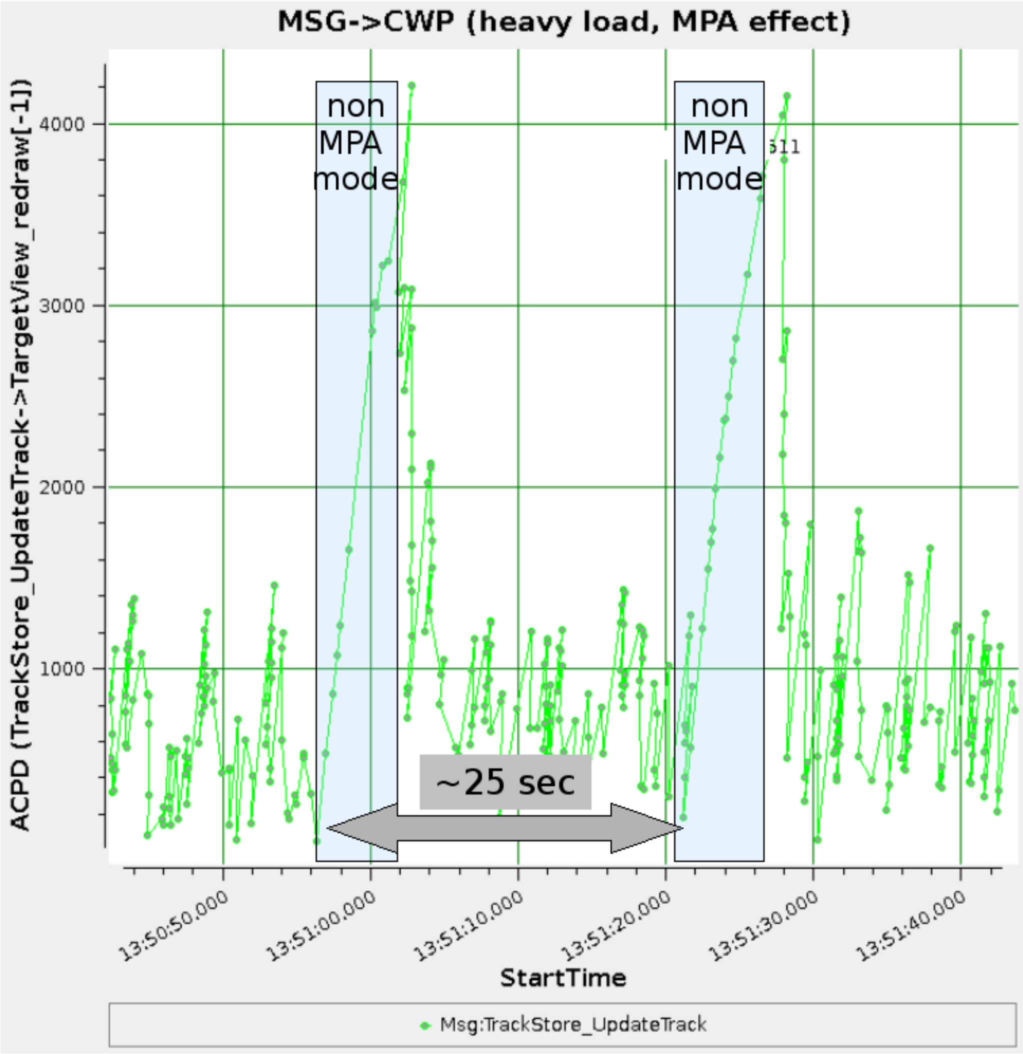
Results: artifical scenario, standalone test bed (1)

- mean response times under heavy load around ~1 sec.
- periodic outlier of ~4 sec.
- bottleneck is CPU of CWP



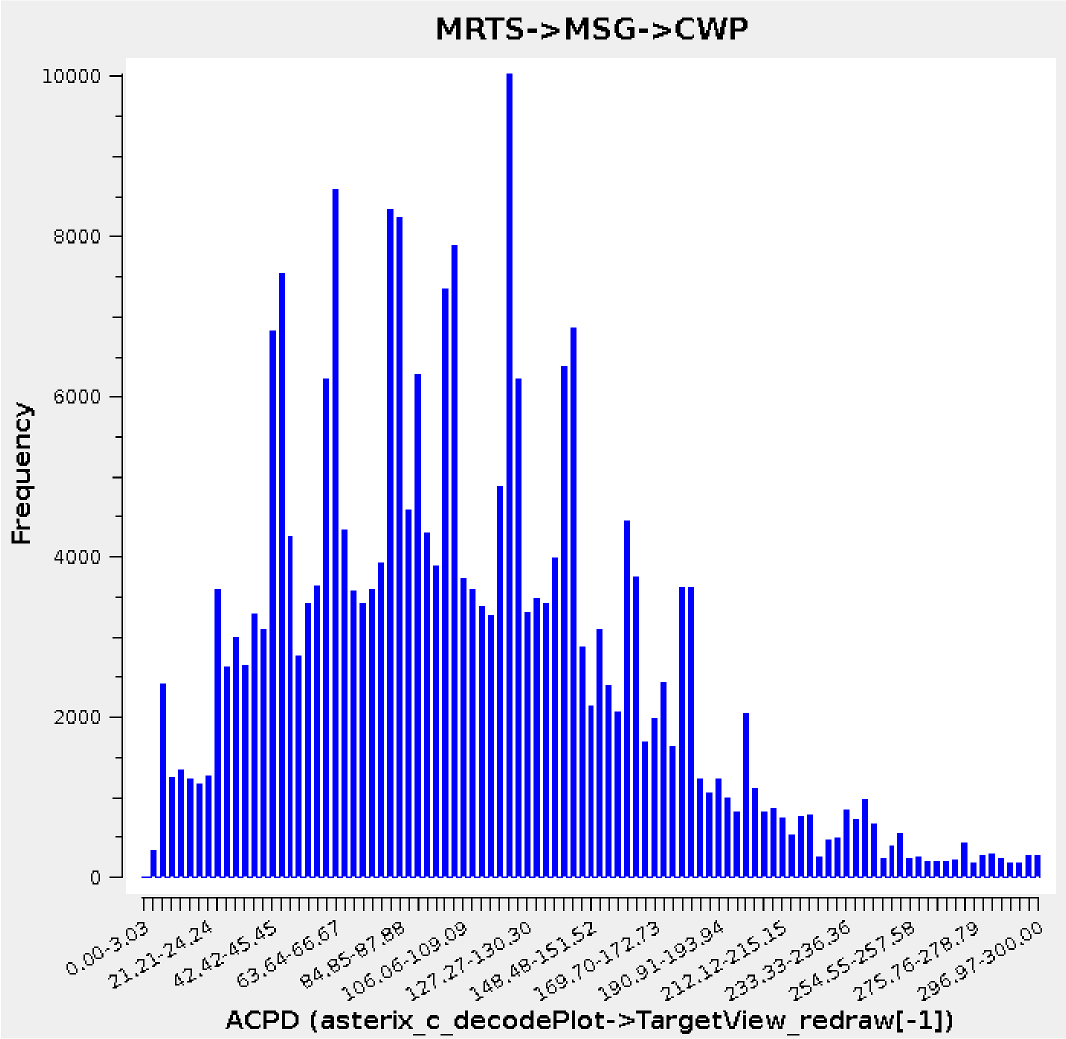
Results: artifical scenario, standalone test bed (2)

- MPA algorithm reduces response times by a factor of 4
- MPA works as designed (reduce CPU usage)
- MPA mode change decision needs to be checked



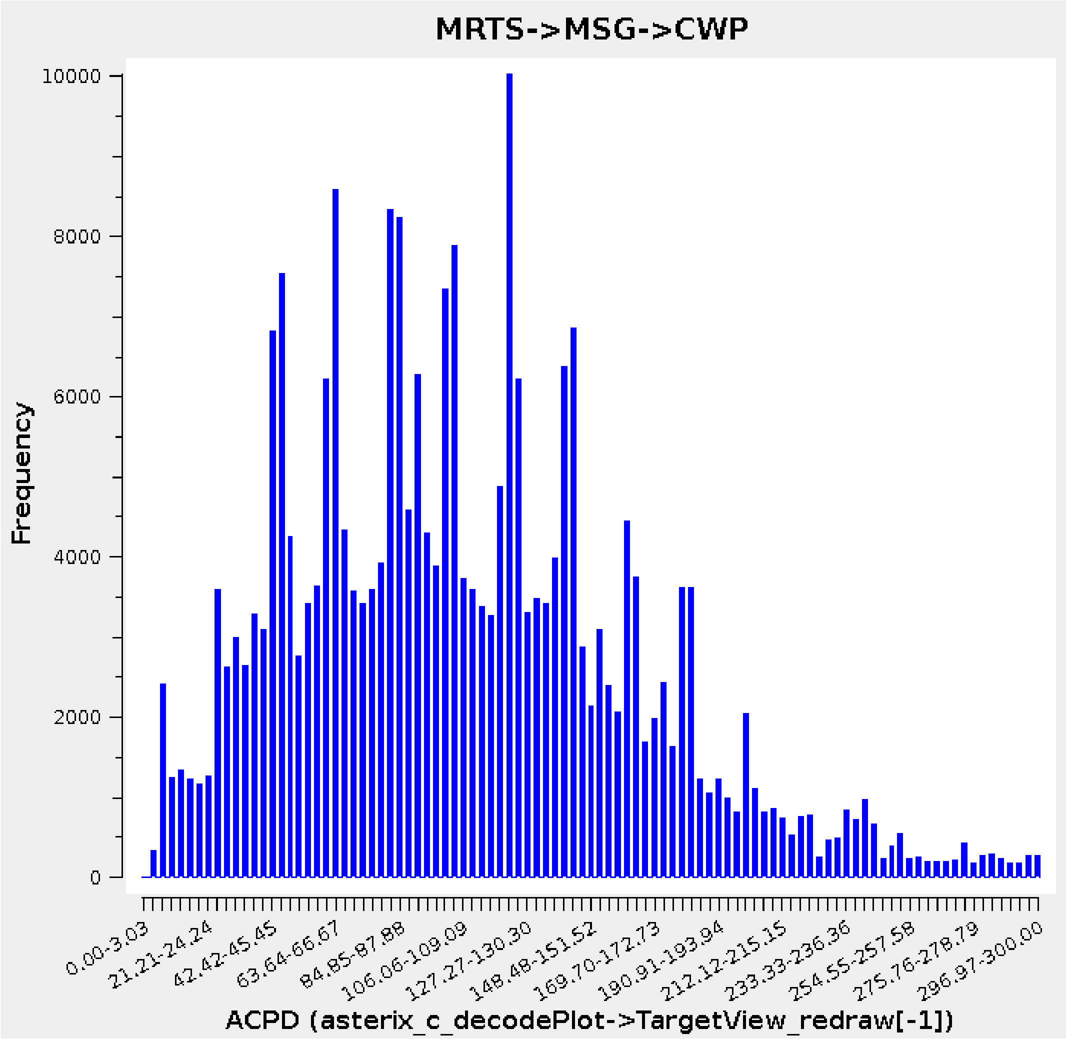
Results: live scenario, distributed test bed (1)

- Mean response time ~137ms
- ~7 times better than 1 second
- In MPA mode much higher mean ~373ms duration (factor 3)



Results: live scenario, distributed test bed (2)

- less than 0,3% of all measurements are above 1 second
- Fits into 3-sigma interval (99,7%)

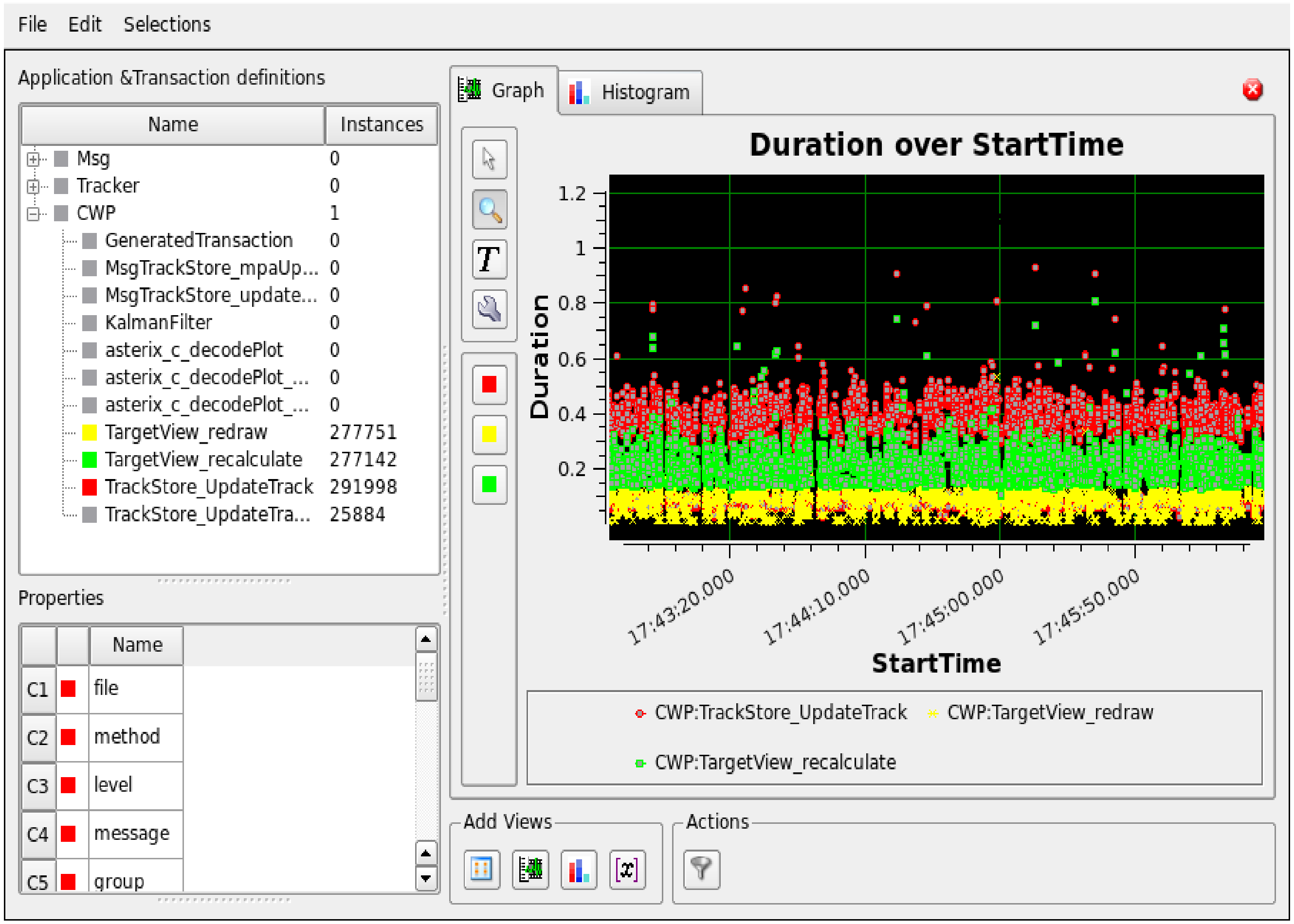


MyARM environment (1)

- Modular design of data sink and data sources for measurement data (DBs: MySQL, Oracle, Postgres)
- Efficient measurement data transport (TCP/IP, Shared-Memory)
- Tools for everyone
  - Command line tools for batch processing
  - Web interface for easy integration into existing platforms
  - Powerful own management application

MyARM environment (2)

- Transaction selection
- Different views of ARM data
- Supports any ARM concepts (Metrics, Properties)



Application &Transaction definitions

Name	Instances
⊞ ▨ Msg	0
⊞ ▨ Tracker	0
⊟ ▨ CWP	1
▨ GeneratedTransaction	0
▨ MsgTrackStore_mpaUp...	0
▨ MsgTrackStore_update...	0
▨ KalmanFilter	0
▨ asterix_c_decodePlot	0
▨ asterix_c_decodePlot_...	0
▨ asterix_c_decodePlot_...	0
🟨 TargetView_redraw	277751
🟩 TargetView_recalculate	277142
🟥 TrackStore_UpdateTrack	291998
▨ TrackStore_UpdateTra...	25884

Properties

		Name
C1	🟥	file
C2	🟥	method
C3	🟥	level
C4	🟥	message
C5	🟥	group

Summary (Results)

- Operational environment (1 sec. requirement)
  - mean response time is about 7 times better
  - response time for presenting a track fits into 3-sigma interval (99,7%).
- Under heavy load (3000 tracks)
  - the mean response times go up to ~1048ms.
  - High peaks (4 sec.) caused by CWP-MPA algorithm

Summary (ARM)

- ARM measurements provide insight view of performance of main processing steps of PHOENIX
- Constant usage of ARM is planned in the software development process of PHOENIX
- Semi-automatic instrumentation of applications would be helpful (Qt® MOC Compiler using QArm)
- ARM can provide more and better information about operational systems to the responsible system management

The end

Thank you for listening

Any questions?