Dr. Gerolf Ziegenhain, DFS GmbH

In Memory of my Boss and Mentor

Andreas Winkelmann

Using Linux in Air Traffic Control

Hardware and Operating System Platforms

Agenda

1. Air Traffic Control for Dummies
2. History of Linux in DFS
3. Challenges, Do’s and Don’ts
4. Modular Platforms Today
5. What’s next?


C++ FOR DUMMIES
ASP.NET 3.5 FOR DUMMIES
eBay PowerSeller Business Practices For Dummies
Harvey
Excel FOR DUMMIES
Harvey
Excel 2003 FOR DUMMIES
Harvey
Excel 2007 FOR DUMMIES
8 BOOKS IN 1
Harvey
2007 DESK REFERENCE FOR DUMMIES
8 BOOKS IN 1
2007 DESK REFERENCE FOR DUMMIES
FOR DUMMIES


114

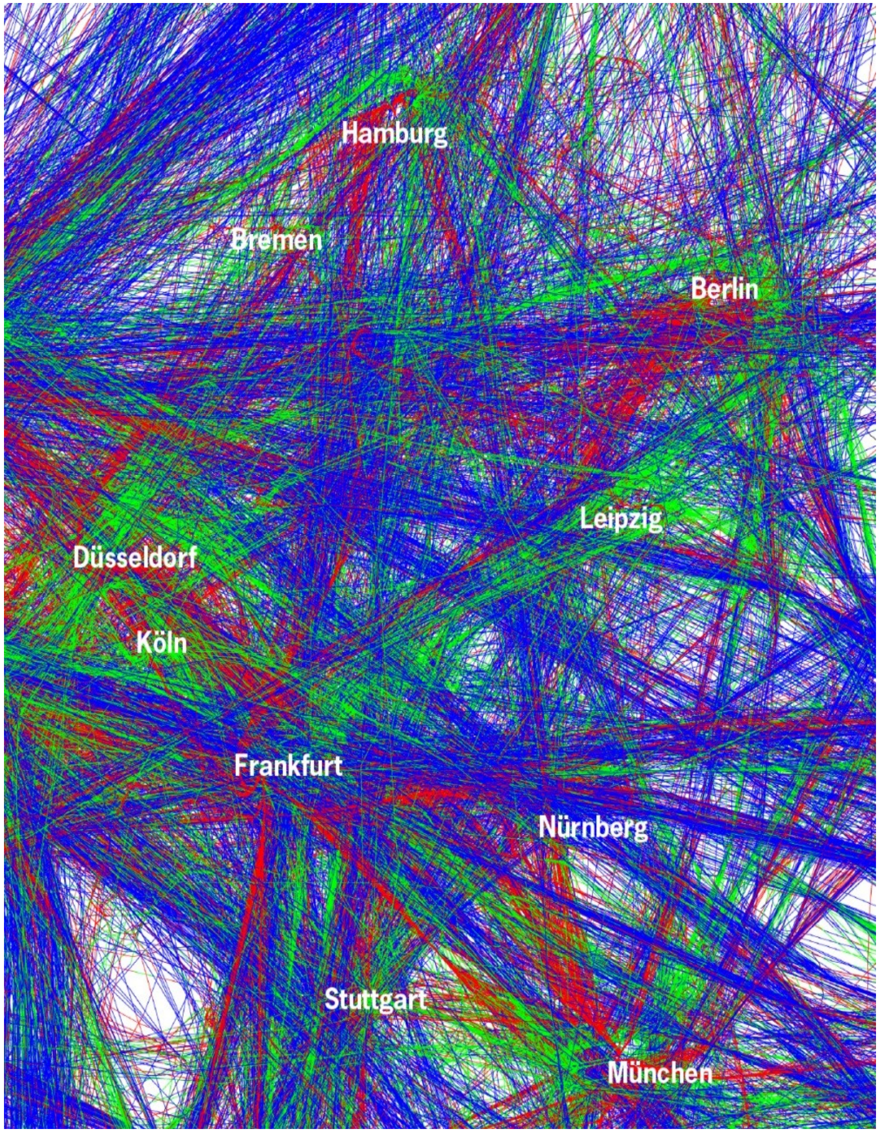
Flight Movements - Make Them Safe & Efficient!

1 day in january 2014

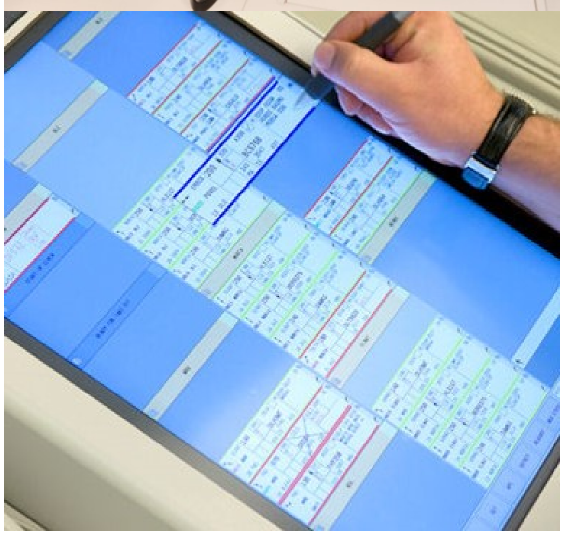
Climb

Upper Air Space

Descent



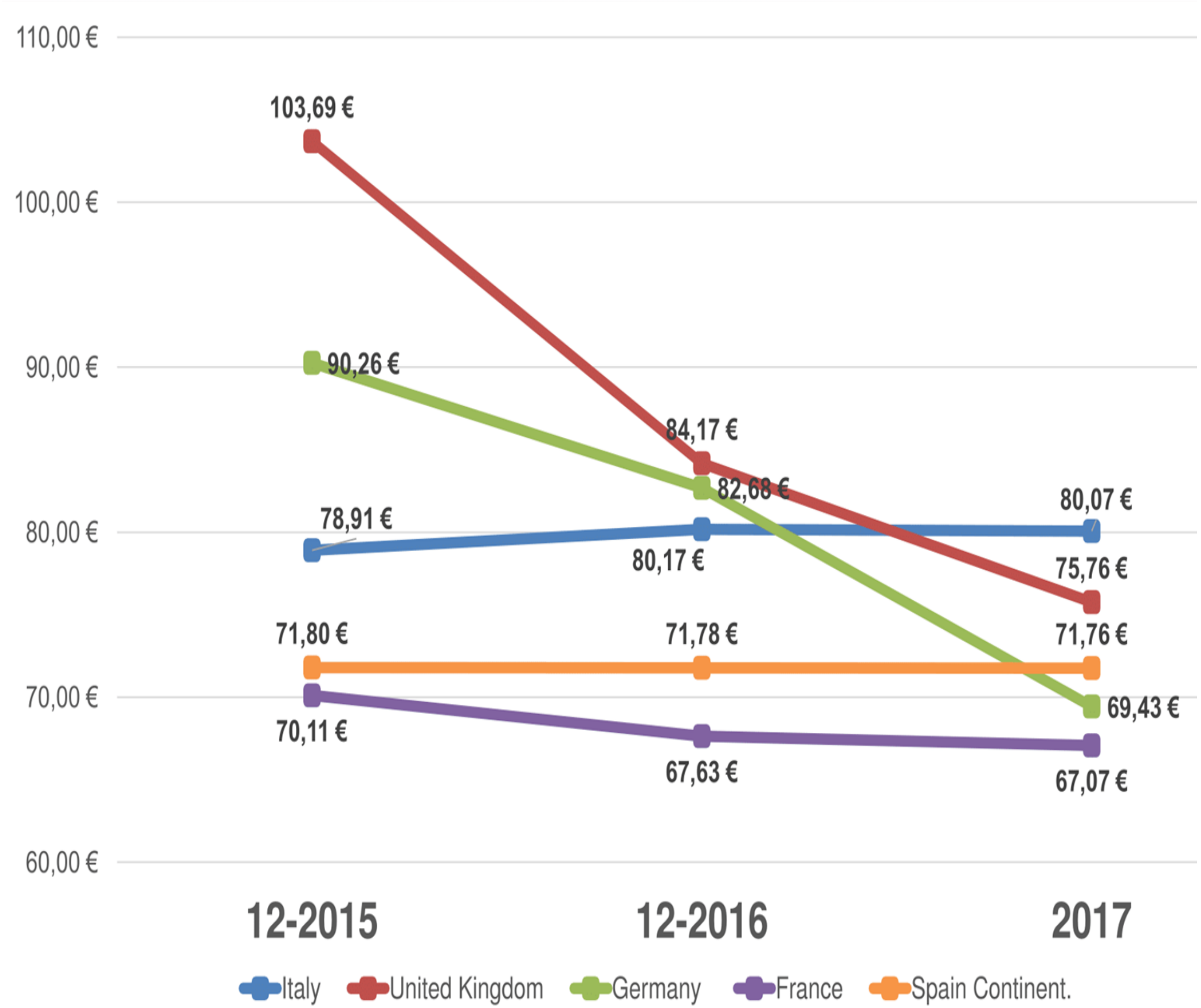
We Design, Build and Operate Infrastructure



Examples of (Integration-)Requirements

- Full Support during the Lifecycle of the Application
- High quality standards, regulations (→ ESARR 6, ED 153a, ...)
- Proofs of factory acceptance tests (paperwork alone is insufficient)
- No Single Point of Failure
- Signals for Switchover to Fallback System

Environment: EU Regulations ⇒ DFS



Unit Rates 2015-2017 (Big 5)

- FABEC?
- SESAR?
- Free market?
- Costs?

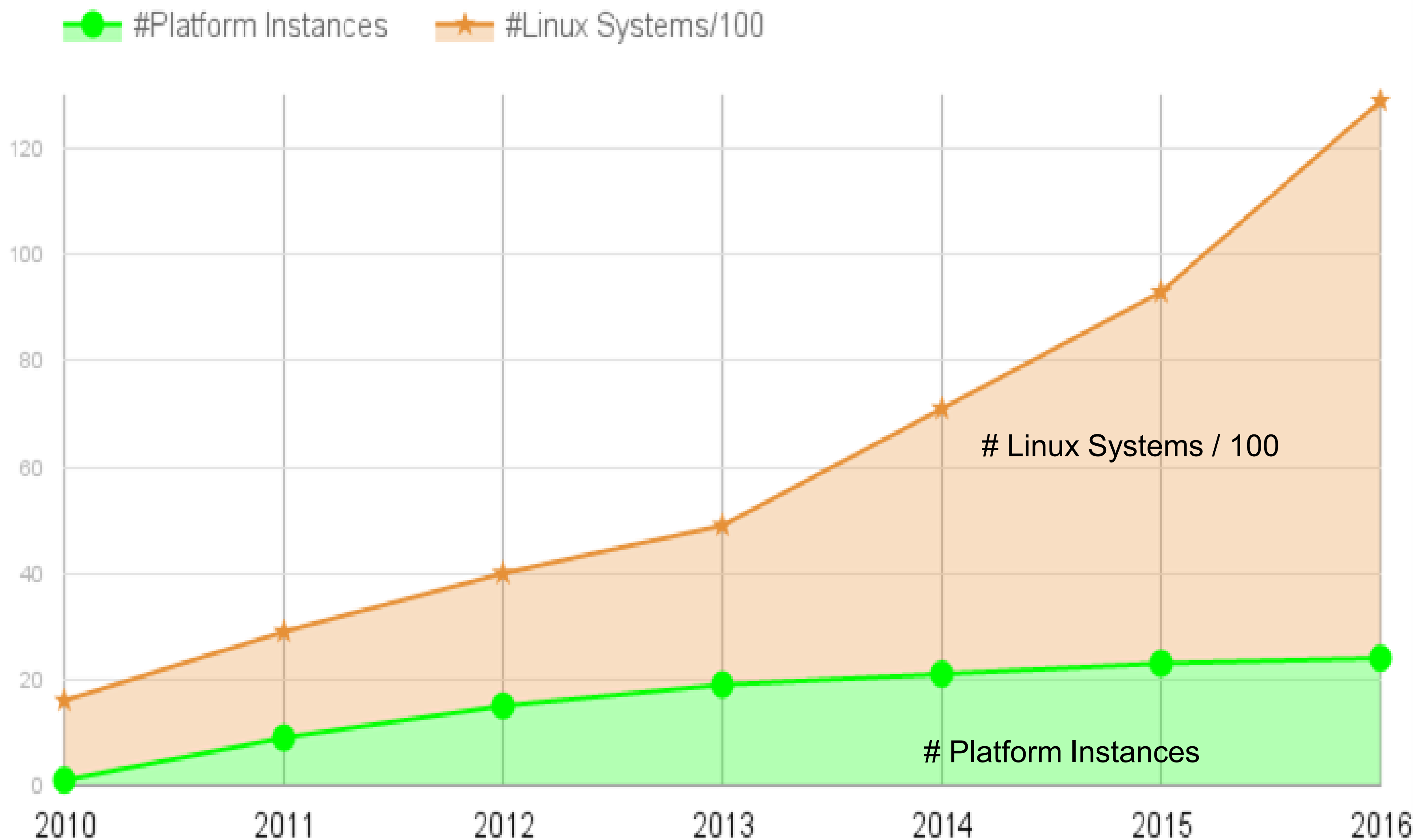
How can Open Source Software and safety critical systems work together?

History of Linux in DFS

History of Linux in DFS

2005	Foundation of LCSC Partnership with SUSE and Red Hat	Centralized Linux Know How Core business ← Data center
2008	Custom DFS Solution: “DFS Linux”	Implement Requirements only once
2010	Linux Platform Hardware & Linux Standard	Harmonize Configurations Harmonize Server Hardware 1.500 Linux Systems
2015	Hardware Platform	Extend HW Standard to Plattform
2016	Linux Platform Re-Focussed	Configurations by customer 10.500 Linux Systems

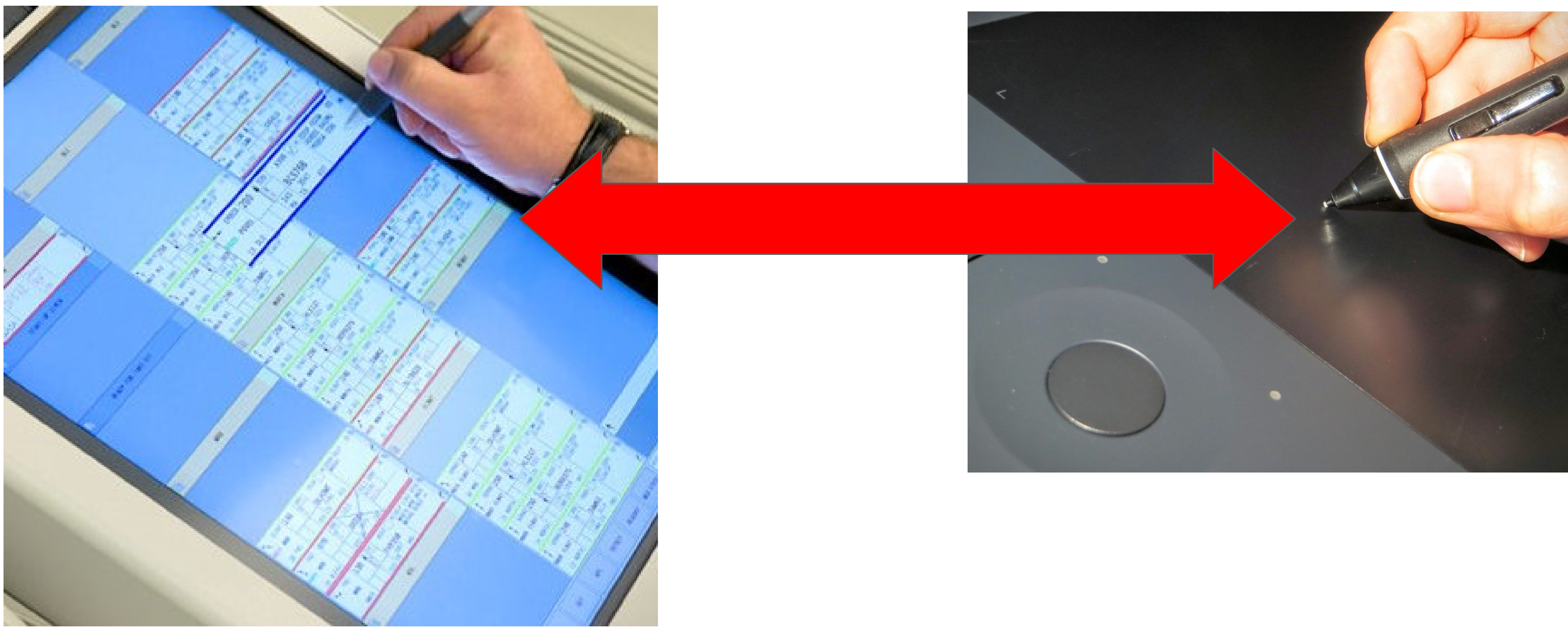
Success Platform at DFS



Challenges, Do’s and Don’ts

CT1: Wacom Driver

Challenge	**Do**	**Don’t**
Wacom Graphics Tablet in a special use case	• Use Customer Test • Plan ahead for support	Wonder if standard driver does not work



CT2: iAMT ACUConfig

Challenge

- HW Bought 2014
- Go Live in 2017
- ACUConfig support not for current Linux in 2016

Do

- Use Open Source Tools
- Assert Support for complete lifecycle
- Have alternatives

Don’t

Rely on binary tools, even if delivered for free

CT3: Deployment Solution

Challenge

- ... need a deployment system.
- Existing customer process uses images.
- We know the state-of-the-art way!

Do

+ Value @ customer
- Cost @ LCSC

⇒ Focus: solving customer problem!

Don’t

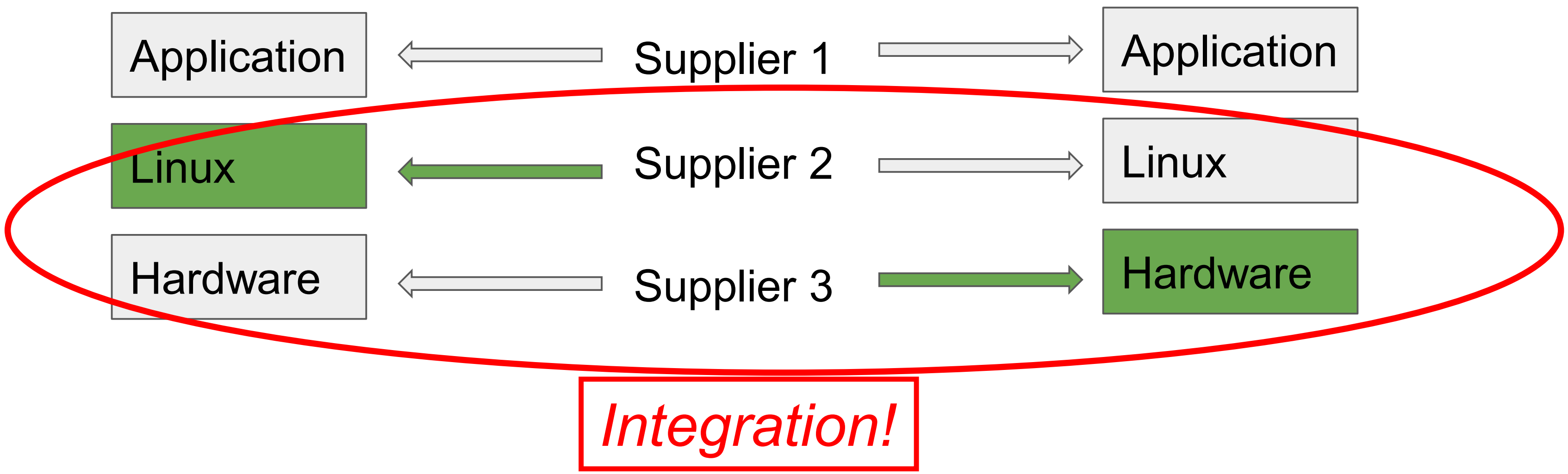
- Use Puppet master
- Live reconfiguration
- … use all the nice features

C1: Gap in Delivery

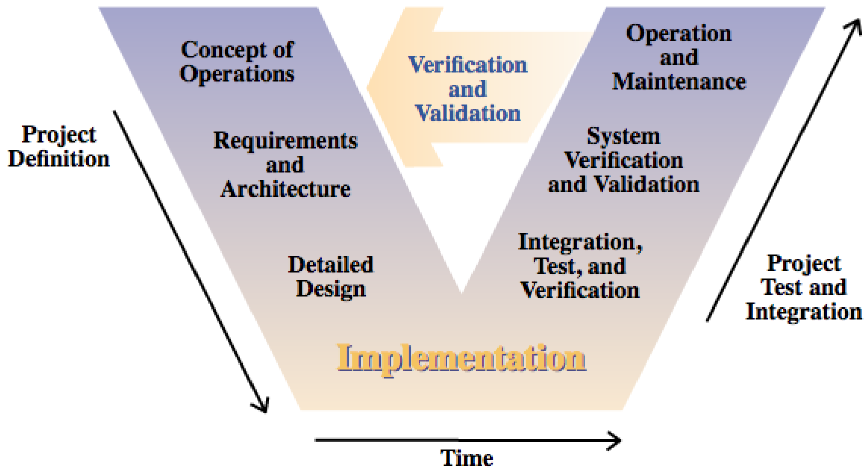
Challenge - What is in the gap?

For a specific Linux Version

1. HW & FW Version for HW Unit
2. Driver for HW Modules
3. Tools for HW functionality



C1: In the V-Model



SSS: HW Unit X by IHV1 shall boot Linux Y.

This test will be done by customer.

SRS: Driver Z for Module V shall boot Linux Y.

This test will be done by IHV.

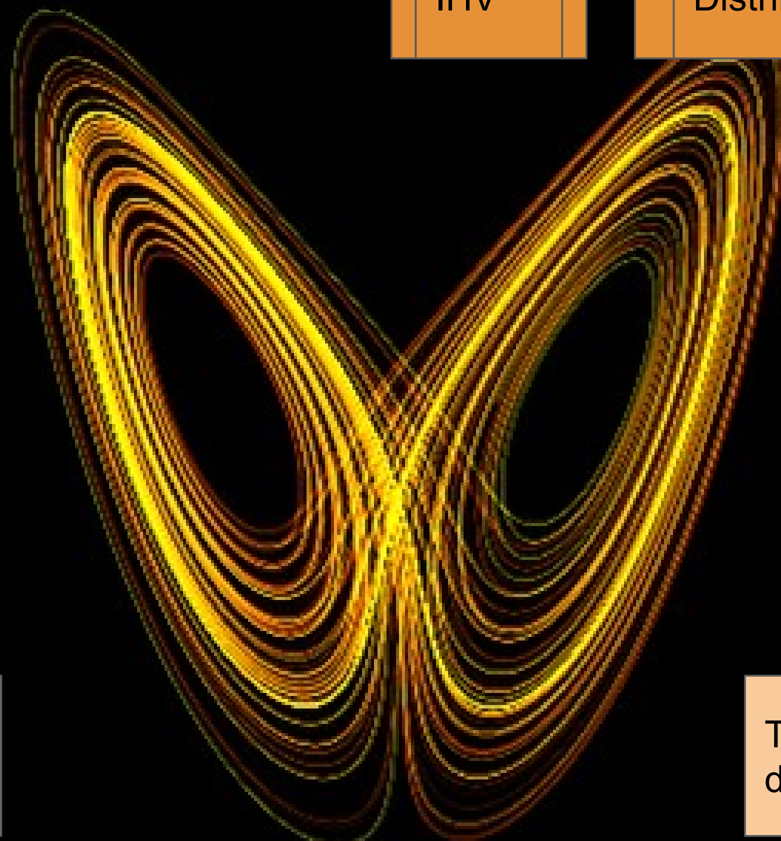
Code: Driver Z for Module V for Linux X boots.

Module-Test: Driver compiles.

Does IHV

Does Distributor

Does Manufacturer X



SSS: HW Unit X by IHV1 shall boot Linux Y.

This test will be done by customer.

SRS: Driver Z for Module V shall boot Linux Y.

This test will be done by IHV.

Code: Driver Z for Module V for Linux X boots.

Module-Test: Driver compiles.

C1: Gap in Delivery

Challenge - What is in the gap?

For a specific Linux Version
1. HW & FW Version of HW Unit
2. Driver for HW Modules
3. Tools for HW functionality

Don't

Rely on (… Examples)
1. HDD order on boot
2. RAID Controller, Network, graphics drivers
3. ACUConfig

Do - in Priority Order

1. Clarify and FAT Integration Requirements (Frame Contracts)
2. Create a HW - OS ICD
3. Demand and steer support
4. Use Developers (In-House and External)

C2: Lifecycle

Challenge

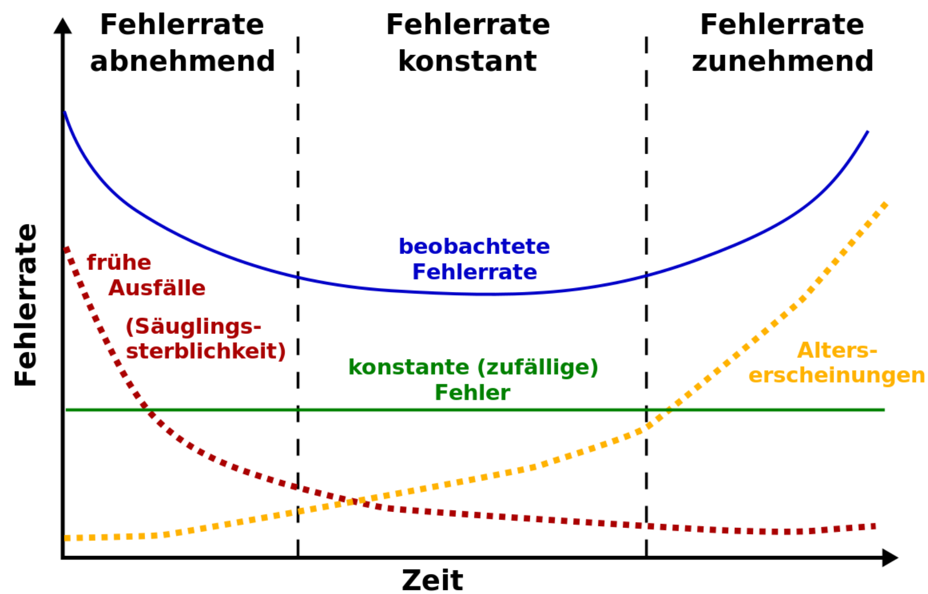
- ATC Application ~8 years
- Safety Requirements
- SL1 & 2 with “drivers license”
- HW with same revision level (HW & FW) only within one order
- OS out of support by default

Don’t

- Rely on manufacturer support

Do

- HW Lifecycle down to revision levels
- Build a modular standard
- In-House support (incl. repairs)
- Know your bathtub-curve



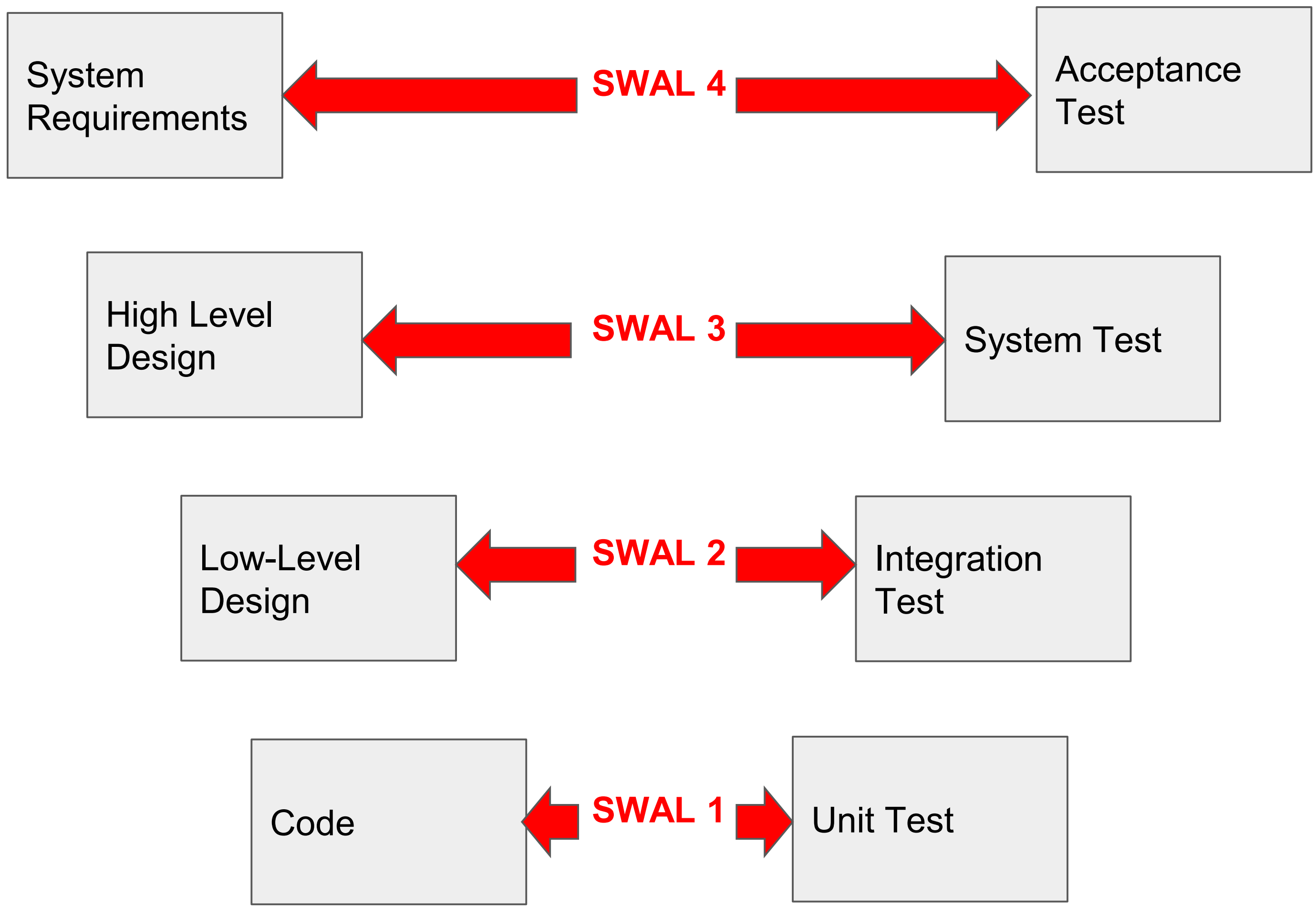
C3: Regulations & Open Source Development

Challenge

DFS:

- Regulations:
  ESARR 6, ED153a, ISO 9001, …
- State-of-the-Art of course…
  ISO 12207, ISO 15288, …
- DFS Core processes:
  “Operating Instructions ATC”

C3: Quick-Intro Software Assurance Levels



C3: Regulations & Open Source Development

Challenge

DFS:

- Regulations:
  ESARR 6, ED153a, ISO 9001, …
- State-of-the-Art of course…
  ISO 12207, ISO 15288, …
- DFS Core processes:
  “Operating Instructions ATC”

Open Source Community:

- If Hans has the display, he will write the driver for you.
- No common development process.

C3: Regulations & Open Source Development

Challenge

DFS:

- Regulations:
  ESARR 6, ED153a, ISO 9001, …
- State-of-the-Art of course…
  ISO 12207, ISO 15288, …
- DFS Core processes:
  "Operating Instructions ATC"

Open Source Community:

- If Hans has the display, he will write the driver for you.
- No common development process.

Distributor:

- Shields by integration of packets

Do

1. Agree on Task & Responsibilities:
   i.e. HW & SW Requirements Specifications, Service Specifications
2. Document processes and interfaces of distributor / maintainer:
   i.e. ESARR 6 Chapter on COTS Software
3. User Acceptance Tests
   i.e. CM lists for hardware, integration with Linux, Regression Testing

Don't

Underestimate the effects of differences in culture and processes.

C4: Customer Acceptance

Challenge

We don't own the production line.
⇒ The customer has to like to use it!


LEGO

C4: Customer Acceptance

Challenge

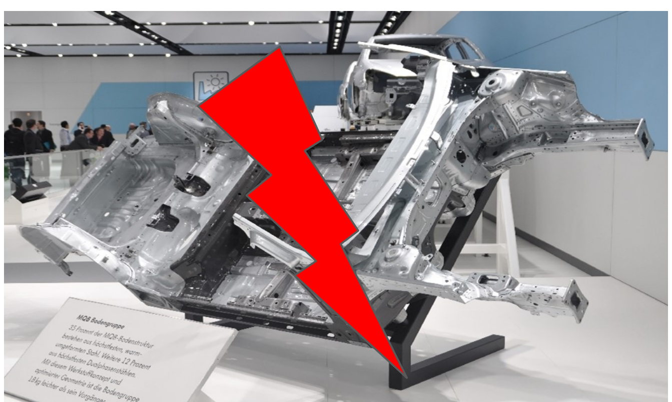
We don’t own the production line.
⇒ The customer has to like to use it!

Don’t

1. Use Force
2. Wrong Vision
3. Know it better
4. Incompatible Process

Do

1. Build flexible, modular Standard
2. Document, KISS, Empower
3. Solve customer problem



C5: Testing

Challenge	Do	Don't
Test-Driven development sounds nice!	Know your customer acceptance test.	Automate as hell.

Modular Platforms Today

DFS Platforms

Hardware Platform
Server, Clients, Displays in frame contracts
Shared common requirements
Modularity, Service

Linux Platform
Enterprise Linux Support (SUSE, Red Hat)
ATC Requirements
Modularity, Integration

Architecture Board
Maintenance of Standards
Governance (→ Exceptions)

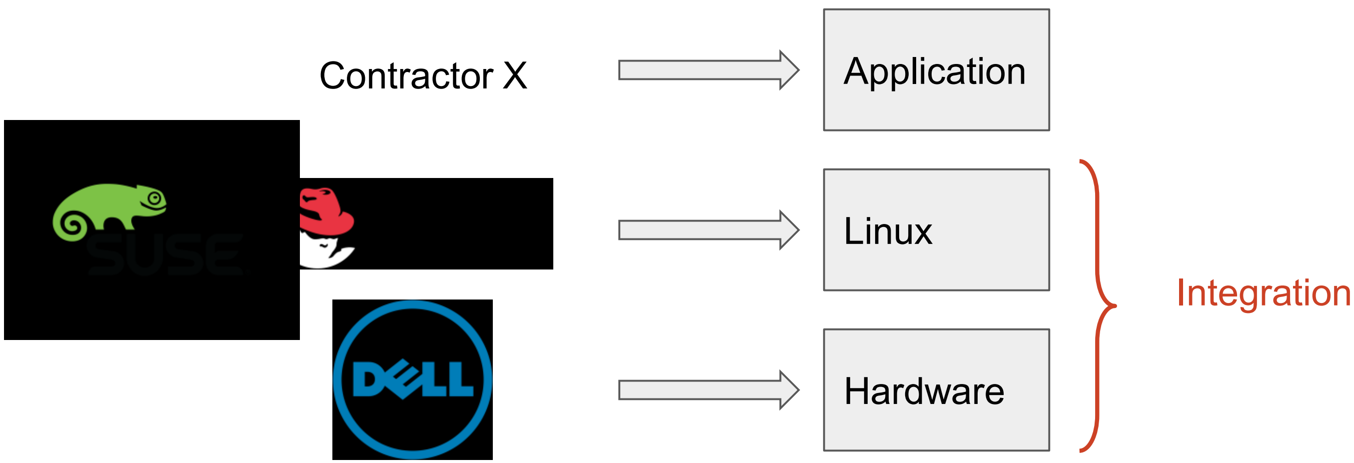
System Architecture Overview

Hardware: HW System = Adaption & HW Unit

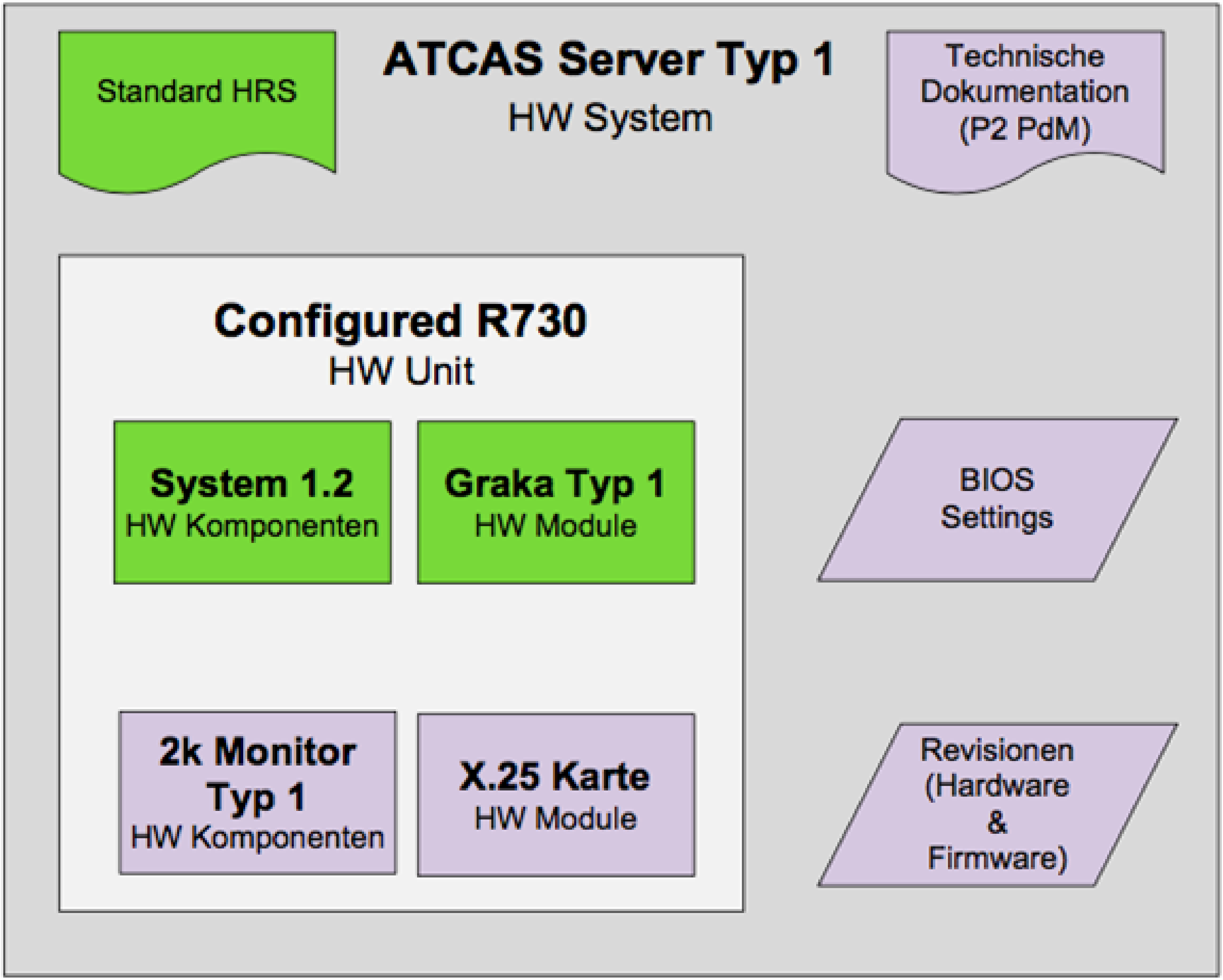
∋ HW Component ∋ HW Module

Linux: OS = Adaption & Distribution

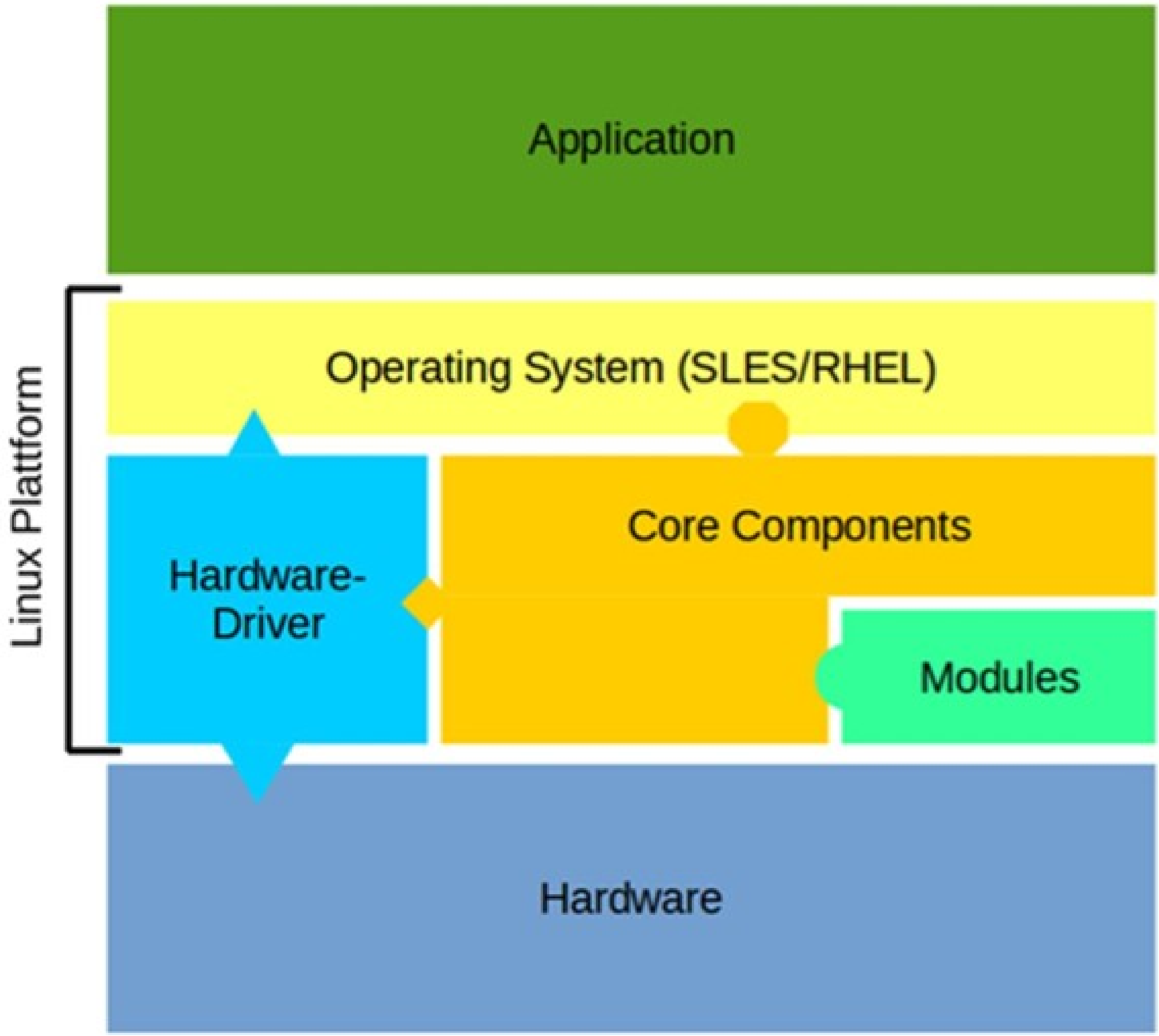
∋ OS Component ∋ OS Module



Architecture of the HW Platform



Architecture of the Linux Platform



What is inside (Core Components / Modules)?

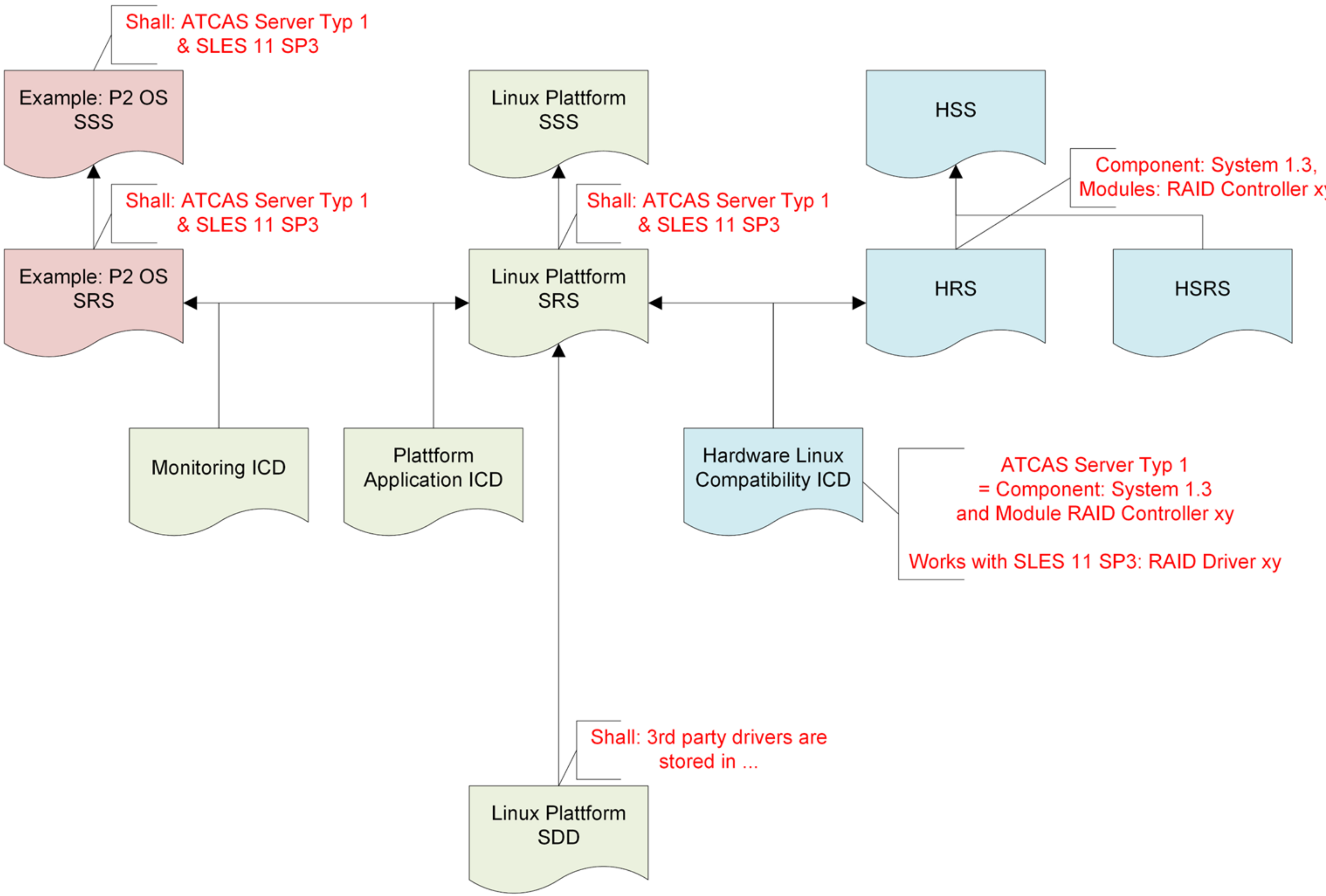
Core Components

1. Stage 1 (Boot & Install)
2. Stage 2 (Configuration)
3. Operational Mode

Modules

1. Deployment
2. Rollback
3. Monit

Platform Requirements Structure (Modular)



What’s Next?

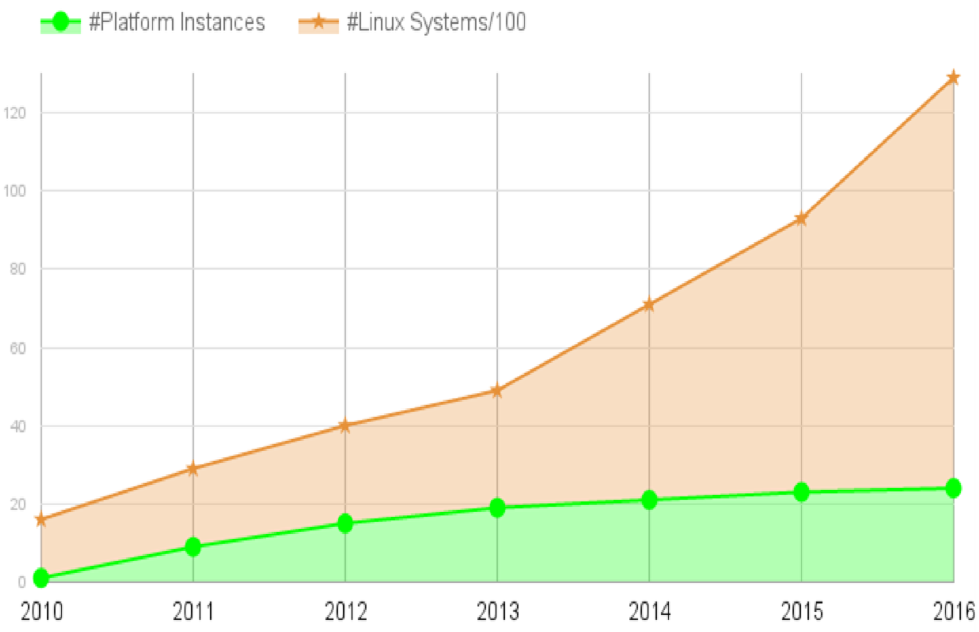
Next Steps

1. OS Platform: continue to shrink the core, empower the users
2. HW Platform: include more HW modules and components
3. Technology: (container) virtualization (SWIM)
4. Regulatory: Security demands

Summary

Summary

+ Modular platform
+ SWAL & Linux
+ Results & Fun



If I understand the business models (**costs**) and **processes** of my **customers** and the Linux Distributors and **focus on core results** it is possible to fulfill the quality requirements of mission critical systems with Linux.