DEVELOPMENT TOOLCHAIN ON STEROIDS

MICHAEL KOLB
THE HISTORY OF THE CLOUD - PART 1

1980: THE PC WAS BORN

... AND THIS IS MY PRIVATE CLOUD. JUST FOR ME!

http://geek-and-poke.com [CC BY 3.0]
About me
Michael Kolb

• Chief Architect for Cloud-Projects @ Robert Bosch in Stuttgart, Germany
• 10 Years+ as Architect and Head of development for IoT & Cloud Projects

#Blogger, #Open Source Contributor, #Cloud Native
About us
Robert Bosch GmbH - Business units

Mobility Solutions
Industrial Technology
Energy and Building Technology
Consumer Goods
E/E Architecture Trend

**TODAY**

**Distributed E/E architecture**

- Modular
- Integration
- Each function has his ECU

**TOMORROW**

**Domain Fusion**

- **Centralization**
- Central Domain ECUs

**Cross** Domain centralized E/E architecture

**FUTURE**

**Vehicle Cloud Computing**

- Vehicle functions in the cloud

**Vehicle centralized E/E architecture**

**(Cross) Domain centralized E/E architecture**

- Central Cross Domain ECUs

**New E/E-Architectures are driving new SW-Architectures**

© Robert Bosch GmbH 2018. All rights reserved, also regarding any disposal, exploitation, reproduction, editing, distribution, as well as in the event of applications for industrial property rights.
About us

Cross Automotive Platform – System, Software, Tools

Software Architecture

Hardware Development

Engineering Services and Solutions

Tools and Methods

Embedded SW

Base Software

Multicore

HTML

C++

Eclipse

Requirements Engineering

Git

Cloud

C

Java

Simulation

Perl

Python

Compiler

OS

XML


© Robert Bosch GmbH 2018. All rights reserved, also regarding any disposal, exploitation, reproduction, editing, distribution, as well as in the event of applications for industrial property rights.
CAP-SST Projects

- Development of a platform with integration of AUTOSAR Adaptive to allow the concept of vehicle computers
- Analysis and introduction of a common language for code generators
- Development of simulation-based solutions for SW testing
- Development of a common cloud based environment for automotive SW development
- Concept coordination and base software development for a BBM-wide controller
- Development of a tooling platform to improve the performance of Multicore SW
- Bosch Mobility common platform
THE GOOD, THE BAD, THE UGLY

LOCAL DEVELOPMENT ENVIRONMENTS
How does the classic local environment look like?

Components

**Datacenter**
- Static Code analysis
- Buildserver
- SCM
- Ticket System
- Artifact Storage

**Local Machine**
- IDE, Compiler, Clients, other Tools
How does the classic local environment look like?

Process

- Requirements
- Code Changes
- Analysis
- Build Output
- Deployment
- System Tests
How does the classic local environment look like?

Developer’s machine

- Code
- Build
- Deploy
- Fix
- Test

IDE settings? Integrations? Quality?
Tools? Configurations? Reproducibility?
Credentials? Access? Environment?

Codebase? Validation?
“Works on my machine” Runtime?
Runtime?
Codebase?
Validation?
HOW DOES CLOUD HELP

ADVANTAGES OF A CLOUD DEV-ENVIRONMENT
Cloud development environments
What can be solved better?

- Uncomplicated to test
- No update hell / version lock-in
- Complete Baseline
- Less effort for Ops
- Pay as you go
- Accessible from everywhere
- “Infinite” computing power

Everything

!(Depending on the perspective)
Cloud development environments

*aaS

Classic

IaaS

PaaS

SaaS

Application & Data

OS / Middleware

Virtualization

Hardware

Your Job

CaaS

- Infrastructure (and OS) are abstracted
- No provider lock
- High portability
- Scalability

Your Job
Deciding for a cloud provider

The question is:

Do you really need to decide for one?

No you don’t, but...
THE TECHNICAL STUFF

ARCHITECTURE. TECHNOLOGIES. PITFALLS.
Cloud development environment @ Bosch

Scope of the technical part

- **Plan**
- **Implement**
- **Store**
- **Build**
- **Analyze**
- **Deliver**

**Scope**

- Ticket System
- IDE
- SCM
- Buildserver
- Static Code analysis
- Artifact Storage

© Robert Bosch GmbH 2018. All rights reserved, also regarding any disposal, exploitation, reproduction, editing, distribution, as well as in the event of applications for industrial property rights.
Cloud development environment @ Bosch

Building Blocks: Static code analysis

- Analysis for each commit
- Local analysis possible
- Sonar as automated reviewer
- Build Breaks on
  - Quality gate fail
  - “Red” Issue increase

Check

Code

Results

Code

Results

Results

18
Cloud development environment @ Bosch

Building Blocks: Buildserver

- Pipeline in Repository
- Zero configuration (except secrets)
- Build contains
  - Compile
  - Analysis
  - Versioning + Tag
  - Unit-Test, I-Test and Sys-Test execution
  - Delivery
- Execution in Docker containers
- Max 10 minutes until you get feedback
Cloud development environment @ Bosch

Building Blocks: Buildserver

Custom Service
- Microservice that consumes “Successful Build events”
- Utilizes REST-API of Ticket System to create links to build artifacts on Nexus

Jenkins Bitbucket Plugin
- Checks BB REST-API for new / changed repositories
- Creates new Build-Plans based on Pipelines in the repository
- Provides Webhook Endpoint to trigger builds
Cloud development environment @ Bosch

Building Blocks: SCM

- Distributed System
- Reviews are done by
  - Jenkins: “Green Build”
  - Sonar: Quality Gate passed, “Red Issues”, Code Coverage
  - Developer: Based on a review guideline → Knowledge sharing
- Only approved Pull-Requests can be merged

**Principle:** “Everything that was approved by Sonar & Jenkins is considered to be valid”
Cloud development environment @ Bosch
Building Blocks: IDE - Terminology

Stack
- Generic recipe for creating a workspace for a specific technology
- Contains one or more references to containers

Factory
- Produces workspaces based on the recepies defined in a stack
- Holds the configuration values for variables in a stack

Workspace
- The instance of a stack
- Defines the environment where IDE actions are executed

Che Server
- Master-Controller of all workspaces
- Serves the Web UI

Machine
- Single container inside a workspace
- “Virtual” Computer
Building Blocks: IDE - Overview

- Web-Based IDE
- "Workspace manager"
- Support for custom “Stacks”
- Workspaces consist of one or more containers
- Support for integration into local IDE
Cloud development environment @ Bosch
Building Blocks: IDE - Workflow

Load Web-UI → Create Stack → Create Factory → Run Factory → Create Workspace → Access Workspace

Shortcut
Cloud development environment @ Bosch

System overview

Manager’s playground

Create WS Link → Code Changes → Deliver → Analyze → Baseline Links

Change Notifications → Review

This is your common baseline
Cloud development environment @ Bosch

Deployment & Release

- Deployment / upgrades done automatically
- Deployed environment is controlled via Git-Tag

Sources

Build

Private DTR

Deployment

Public DTR

Infra as code

Microsoft Azure Cloud West Europe

AKS

Private DTR

Public DTR

Deployment / upgrades done automatically
Deployed environment is controlled via Git-Tag
Cloud development environment @ Bosch

Benchmark

- Decreased build times by factor 3 (average)
  - Parallel Testing, even for System Tests
  - Dynamic scaling of build CPUs
- No dedicated Operations needed anymore
- Cooperation is much more likely to happen
- Main accepted Use cases so far
  - IDE for “smaller” pieces of code (e.g. reviews)
  - Back-in-Time Machine for old Build environments
Cloud development environment @ Bosch
Rumors. Prejudices. Lessons learned.

- Docker can be a problem
- Be provider agnostic wherever possible
- Hold everything inside your repository (Code, Infrastructure, Architecture, Documentation, Build, Configuration, Machines, ...)
  - Everything!
- Do not split Dev and Ops
- Automate your quality measures
THANK YOU

WE KEEP IN TOUCH

MICHAEL.KOLB3@BOSCH.COM

We are hiring!
Visit us at our booth 27