



#### Bachelorarbeit - Sean

Aktueller Stand vom 17.07.2018





# Gliederung

- 1. Projekt-Status
- 2. Kommende Technologien
  - i. Spring-Boot-Data/SQL
  - ii. JpaRepository
  - iii. Scheduler
- 3. Organisatorisches



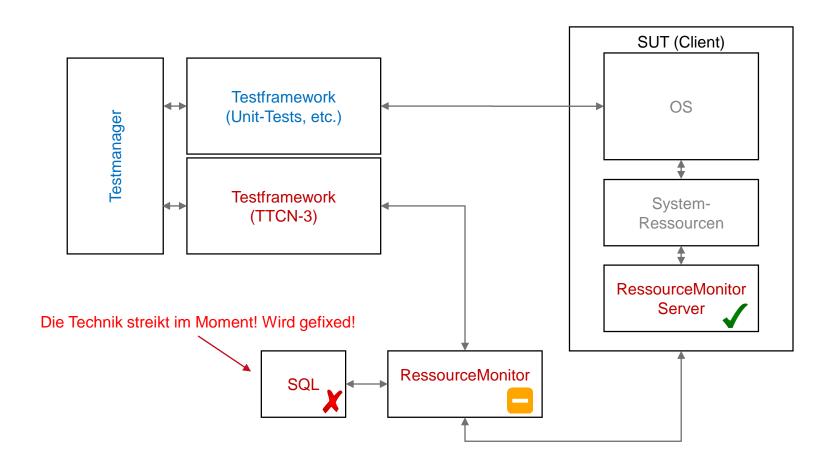


# 1. Projekt-Status





### **Aktueller Status:**







# 2. Kommende Technologien





## **Spring Boot Data**

 Automatische Übersetzung von Klassen zu Tabellen



OneToMany, OneToOne, ManyToOne,
 ManyToMany

```
public class Processor implements Serializable {
    /**
    *
    */
    private static final long serialVersionUID = 1L;
    @Id
    String processorId;
    String processorName;
    String processorVendor;
    String processorFamily;
    String processorFamily;
    String processorModel;
    String processorStepping;
    int processorPhysicalPackageCount;
    int processorPhysicalProcessorCount;
    int processorLogicalProcessorCount;
    int processorLogicalProcessorCount;
    int processorValue> processorValueProcessor")
    Set<ProcessorValue> processorValues;
```





#### SQL-Datenbank

System hostName operatingSystem

hardwareConfig

Measurement measurementld startDate endDate system processes

> HardwareConfig systemModeISN baseboardSN firmwarePK processorId networkMacs memoryTotalspace

Process

name isParentProcess values

process timestamp threadCount priority virtualSize residentSize kernelTime userTime upTime startTime bytesRead bytes Written handles

Process Value

System Model serialNumber manufacturer model

Baseboard serialNumber manufacturer model version

OperatingSystem

manufacturer

family

version

build

codeName

manufacturer description version releaseDate

Firmware

name vendor family model stepping physcialPackageCount physicalProcessorCount logicalProcessorCount isCpu64bit vendorFreq

Processor

ProcessorValue measurementId processorId timestamp systemCpuLoadBetweenTicks systemCpuLoadTicks systemCpuLoad systemLoadAverages processorCpuLoadBetweenTicks processorCpuLoadTicks systemUpTime contextSwitches interrupts

timestamp measurementld networkMac bytes Recv bytes Sent packets Recv packets Sent inErrors outErrors

Network

name

ipv4

ipv6

mtu

speed

values

displayName

NetworkValue MemoryValue totalSpace measurementld timestamp available swapTotal swapUsed

totalSpace

values

Memory

Disk serialNumber model name size fileSystem values

DiskValue diskSerialNumber measurementId timestamp reads readBytes writes writeBytes transferTime

FileSystem FileSystemValues diskSerialNumber measurementId fileStores FileSystemDiskSN timestamp

values

uuid

name

fsType

values

volume

mountPoint

description

measurementld

fileStoreUuid

timestamp

usableSpace

FileStoreValue

totalSpace

openFileDescriptors FileStore maxFileDescriptors





### JpaRepository (Java Persistence API)

- Abstrakte Schicht zwischen Datenbank und Applikation
- Keine differenzierte Implementierung für diverse Datenbanken
- Methodennamen werden in SQL-Anfragen übersetzt

```
count(): long - CrudRepository
delete(Memory arg0): void - CrudRepository
deleteAll(): void - CrudRepository
deleteAll(Iterable<? extends Memory> arg0): void - CrudRepo:
deleteAllInBatch(): void - JpaRepository
deleteByld(Long arg0): void - CrudRepository
deleteInBatch(Iterable<Memory> arg0) : void - JpaRepository
equals(Object arg0): boolean - Object
existsByld(Long arg0): boolean - CrudRepository
findAll(): List<Memory> - JpaRepository
findAll(Example < S> arg0): List < S> - JpaRepository
findAll(Pageable arg0): Page<Memory> - PagingAndSortingR
findAll(Sort arg0): List<Memory> - JpaRepository
findAll(Example<S> arg0, Pageable arg1): Page<S> - QueryBy
findAll(Example<S> arg0, Sort arg1): List<S> - JpaRepository
findAllById(Iterable<Long> arg0): List<Memory> - JpaReposit
findByld(Long arg0): Optional < Memory> - CrudRepository
flush(): void - JpaRepository
getClass(): Class<?> - Object

    getOne(Long arg0): Memory - JpaRepository

hashCode(): int - Object
notify(): void - Object
notifyAll(): void - Object

♠ save(S arg0): S - CrudRepository

SaveAll(Iterable<S> arg0): List<S> - JpaRepository
A caveAndFluch(S aran) · S - InaRenocitor
```

```
public interface MemoryRepo extends JpaRepository<Memory, Long> {
}
```





#### Scheduler

- Feste Scheduler für diverse Messungen
- isRunning = true → Messung wird geupdated
- isRunning = false → iteriere zur nächsten Messung

```
@Component
public class ScheduledTask {
    @Scheduled(fixedRate = 1000)
    public void execute() {
```

LIVE-DEMO





# 3. Organisatorisches

Betreuung, Evaluation "Home Office", Visio-Lizenz