

## Automatisiertes Testen eines Mesh Funk Systems mit echter Hardware

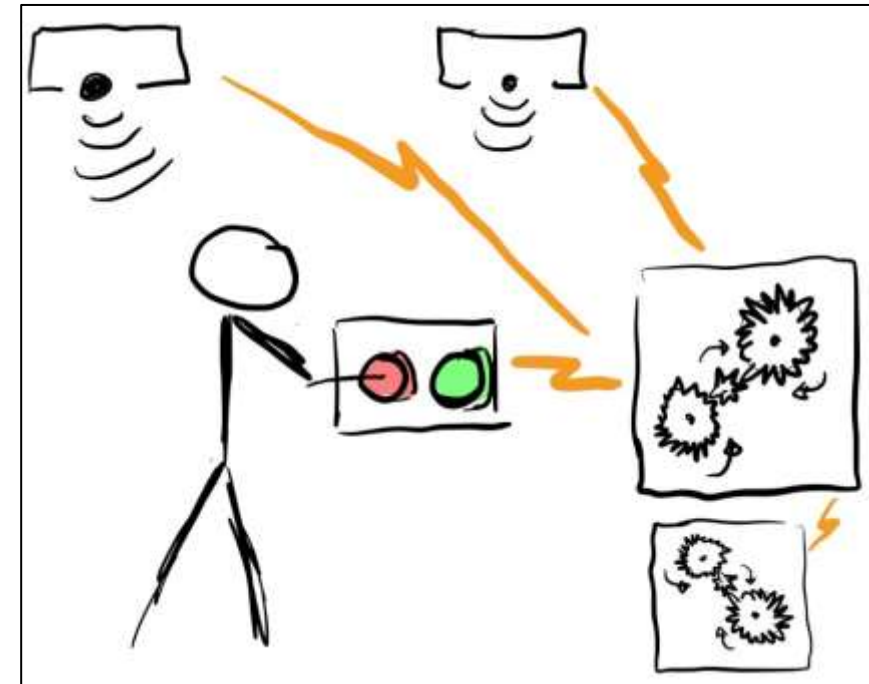
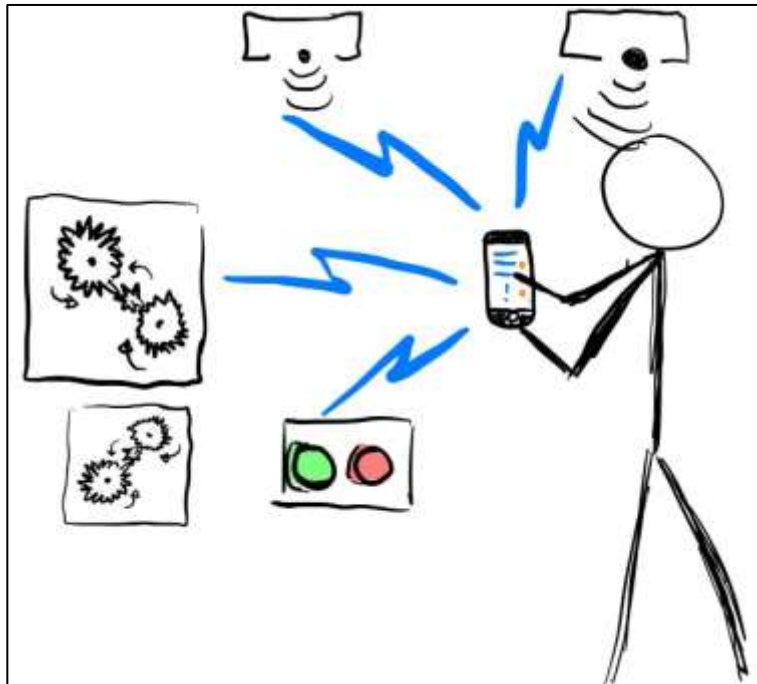


## Inhalt

- Projektübersicht
  - Vernetzte Funkmodul-Firmwareplattform
- Wie testen?
  - Unit-Tests
  - Manuelle Tests
  - Automatisierte Tests mit echter Hardware?
- Hardware Aufbau
- Features
- Repositories & Submodules
- Jenkins Pipelines
- Kosten-Nutzen

## Vernetzte Funkmodul-Firmwareplattform

- Bluetooth Low Energy für Setup/Konfiguration
- Proprietäres Funkprotokoll für Kommunikation im Funknetz
- Parallelbetrieb beider Technologien (Zeitmultiplex)



## Vernetzte Funkmodul-Firmwareplattform

- Einsatz in verschiedenen Sensor-/Aktor-Komponenten
  - Single Chip
  - Dual-Chip (zweiter Mikrokontroller)
  - Dual-Chip (Linux SoC)
- Verschiedene Firmware-Varianten des Host-Controllers
- Verschiedene Firmware-Varianten des Funkmoduls

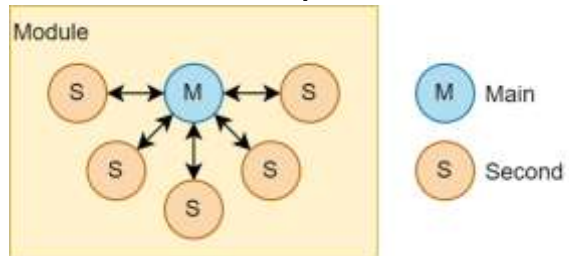


## Vernetzte Funkmodul-Firmwareplattform

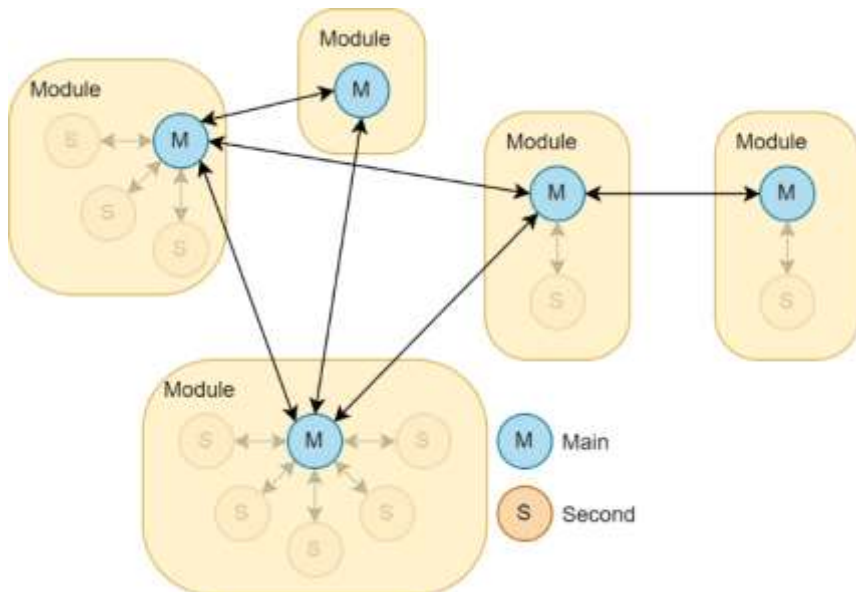
Mikrokontroller	Funkmodul-Firmware	Host-Firmware
nRF52832	RfAktorABC	HostAktorA
		HostAktorB
		HostAktorC
nRF52832	RfAktorD	-
nRF52832	RfPresence	-
nRF52805	RfSensor	-
nRF52840	RfGateway	HostGateway

## Systemaufbau

- Mehrere Komponenten → Modul



- Mehrere Module → Netzwerk

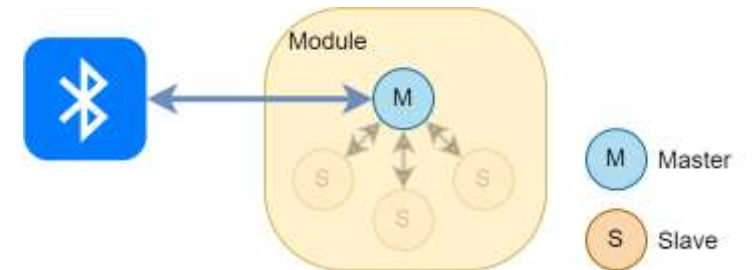


## System-Limitationen

- Pro Netzwerk
  - Maximal 50 Module
- Pro Modul
  - Maximal 16 Komponenten
- Präsenzerkennung und Gateway
  - Bilden jeweils ein eigenes Modul
  - Nur einmal pro Netzwerk

## Konfigurationsablauf eines Moduls

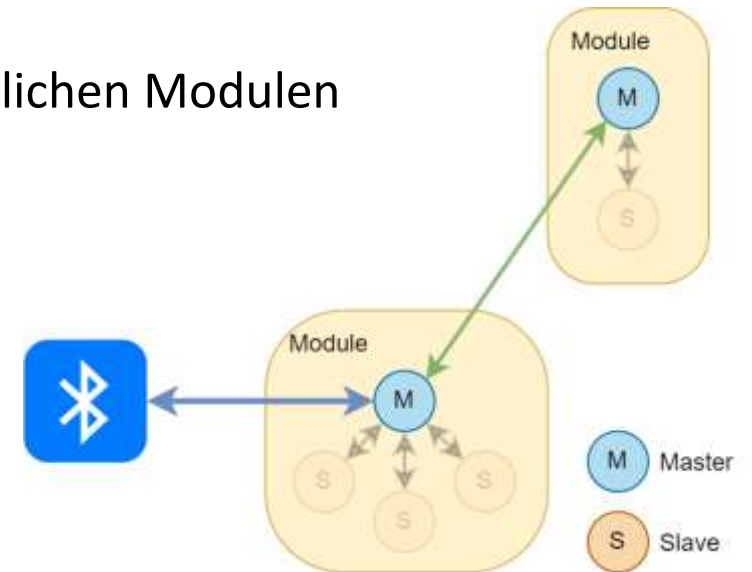
- Bluetooth Advertising wird an den Komponenten aktiviert
- Bluetooth verbinden
- Bildet Technologie-Umsetzer zu restlichen Komponenten des Moduls
- Individueller Schlüssel für jede Funkstrecke im Modul
  - Main zu den jeweiligen Seconds





## Konfiguration eines Netzwerks

- Selber Ablauf wie für Konfiguration eines Moduls
- Bestehender Main bildet Technologie-Umsetzer zu neuem Main im Netzwerk
- Neue Main-Komponente wird in bestehende Installation integriert
- Individueller Schlüssel für jede Funkstrecke
  - Zwischen allen Modul-Mains
  - Keine Kommunikation zwischen Seconds aus unterschiedlichen Modulen



## Wie testen – mit Unit-Tests?

- Einfach umsetzbar für einfach Codemodule
- Aufwändig für hardwarenahen Code
- Aufwändig für Testen von Zusammenspiel mehrerer Codemodule
- Zeitmultiplex der Funkprotokolle nicht testbar

## Wie testen – manuelle Tests?

- Zeitaufwändige Tests
  - Wiederholung von Tests für jede Firmwarevariante
- Konfiguration von grösseren Netzwerken aufwändig und fehleranfällig
- Komplexe Abläufe schwierig nachzuvollziehen
- Schlechte Reproduzierbarkeit von Fehlerfällen
- Nebeneffekte werden bei grösseren Netzwerken übersehen
- Reale User Experience ist nachvollziehbar

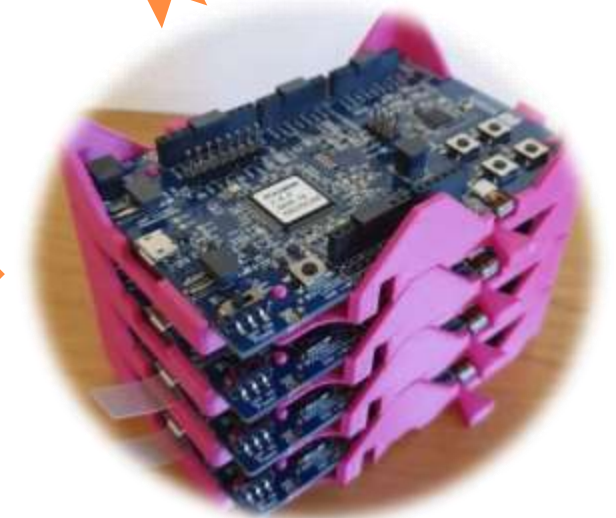
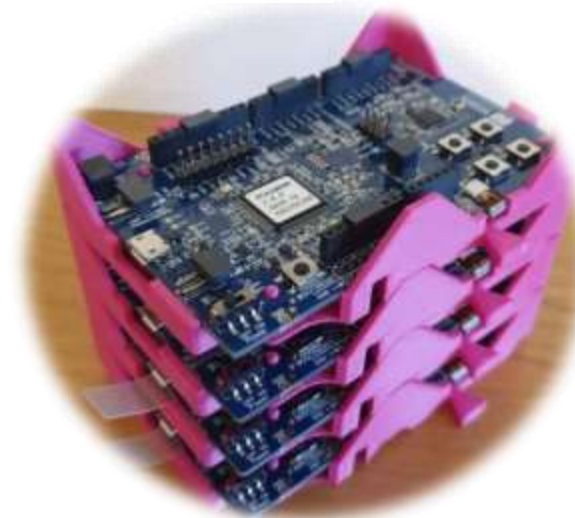
## Wie testen – automatisiert mit echter Hardware?

- Ansatz automatisiertes Testen mit echter Hardware
  - Alle Funkmodule sind Nordic nRF52
  - Mehrere Komponenten nutzen eine UART zum Host-Controller
  - Problematik von schlechten Möglichkeiten der Hardware-Emulation entfällt
  
- Isolation des Funkmoduls für das Testing
  - Erweiterung an UART-Schnittstelle für Stimulation und Auswertung
  - Verwendung von Nordic nRF52 Development kits («einfacher» Hardware-Aufbau)
  - Mock für Host-Kommunikation
  - Verwendung von Nordic nRF52 Development kit anstatt SmartPhone

Hardware Aufbau


**Achtung,  
fertig...**

**LOS!**

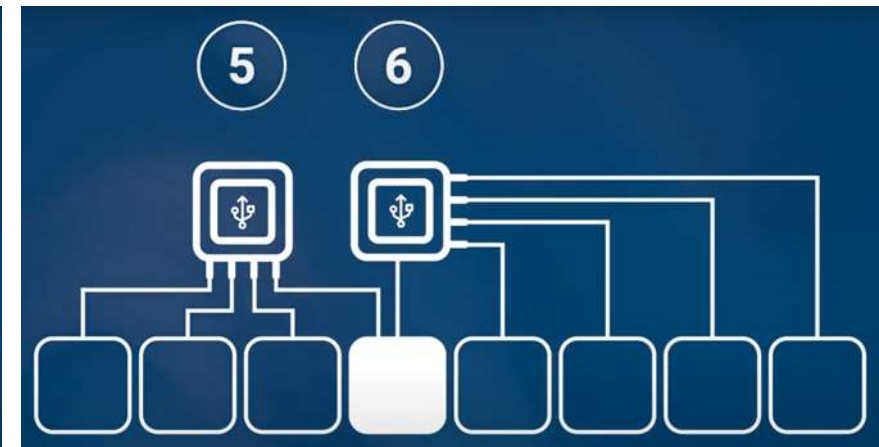
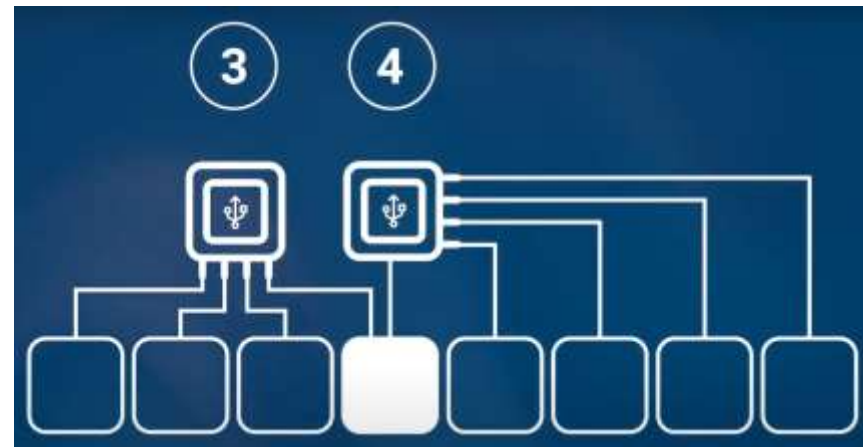
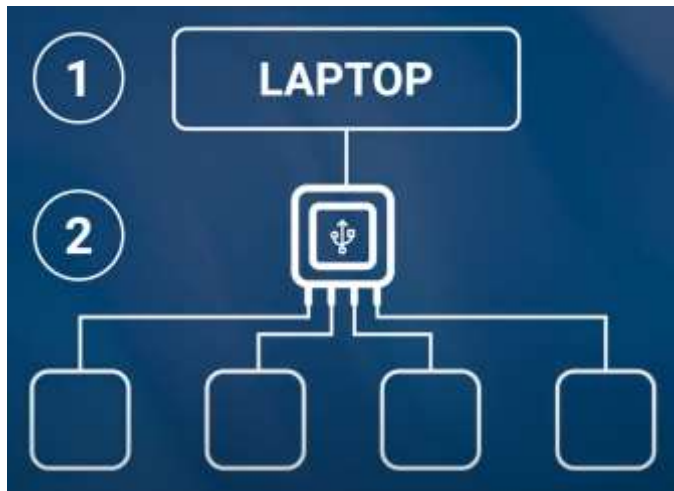


**FAILED**

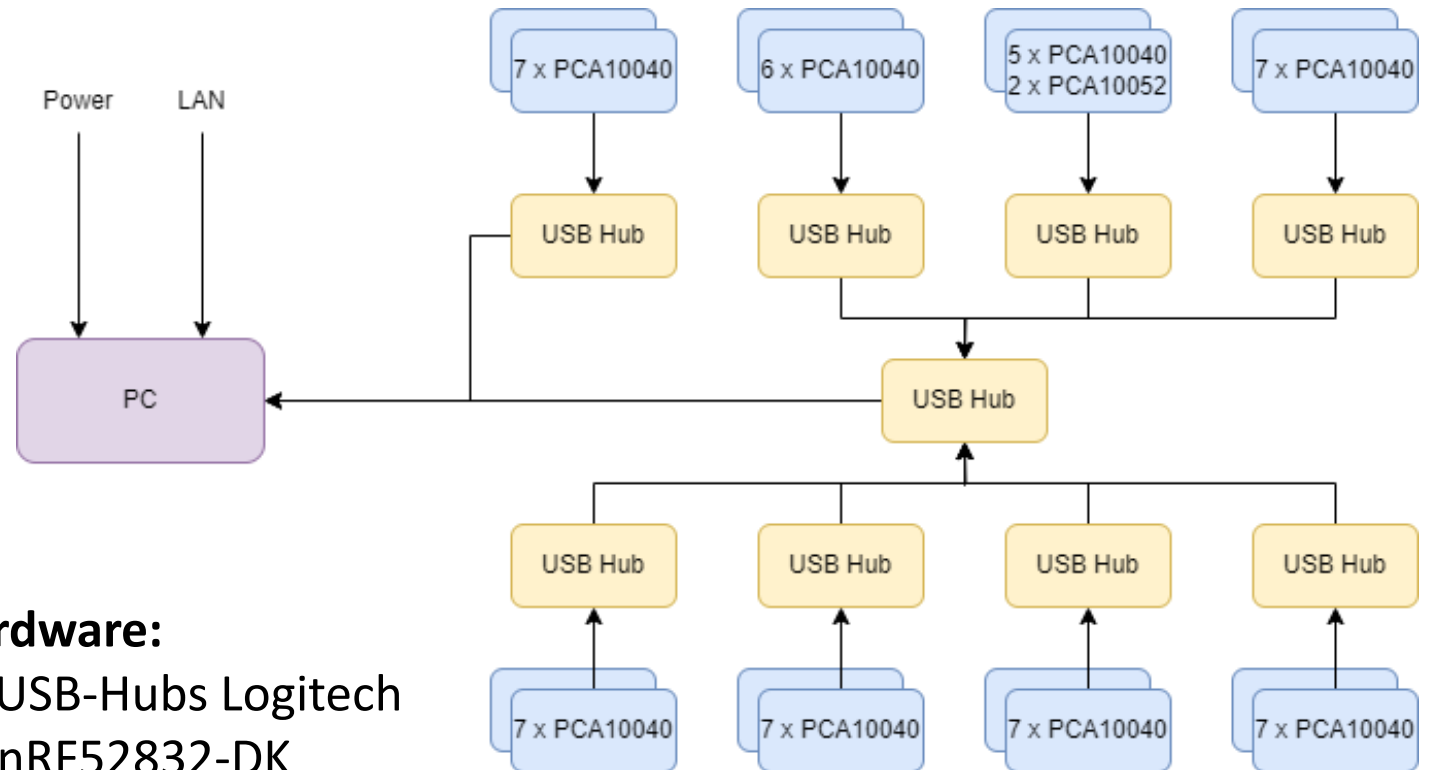
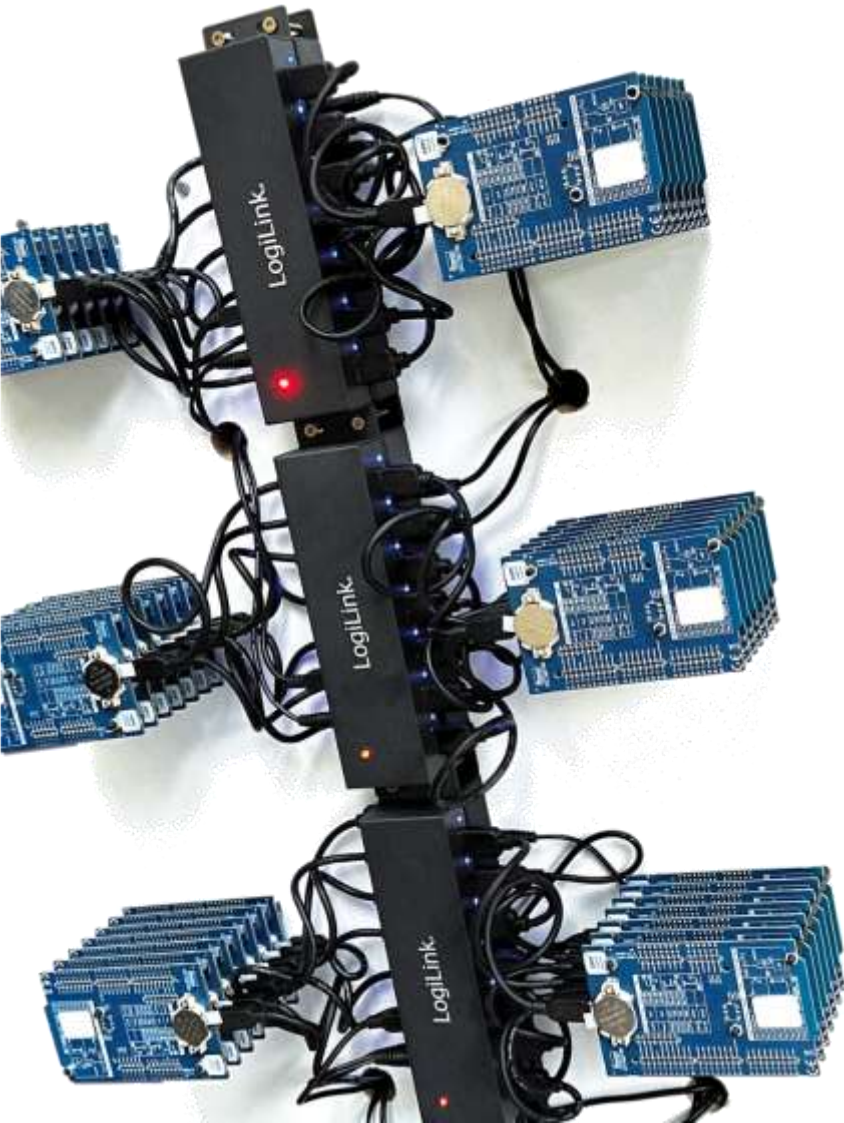
## Hardware Aufbau

- USB-Limitationen
  - Theoretisch maximal 127 Geräte an einem USB-Controller
  - Maximal 7 Hub Levels (Tiers)
  - Die meisten USB-Communication-Chips können nur 4 Ports bedienen
- Hub 1 ist belegt durch den Computer
  - Computer kann auch bereits Hub 2 belegen
- Hub 7 ist das jeweilige Endgerät
- USB-Hubs mit mehr als 7 Ports haben immer weitere Hub Levels intern
-  [USB Device Tree Viewer](#)

## Hardware Aufbau



## Hardware Aufbau



### Hardware:

- 9 USB-Hubs Logitech
- 51 nRF52832-DK
- 3 nRF52840-DK

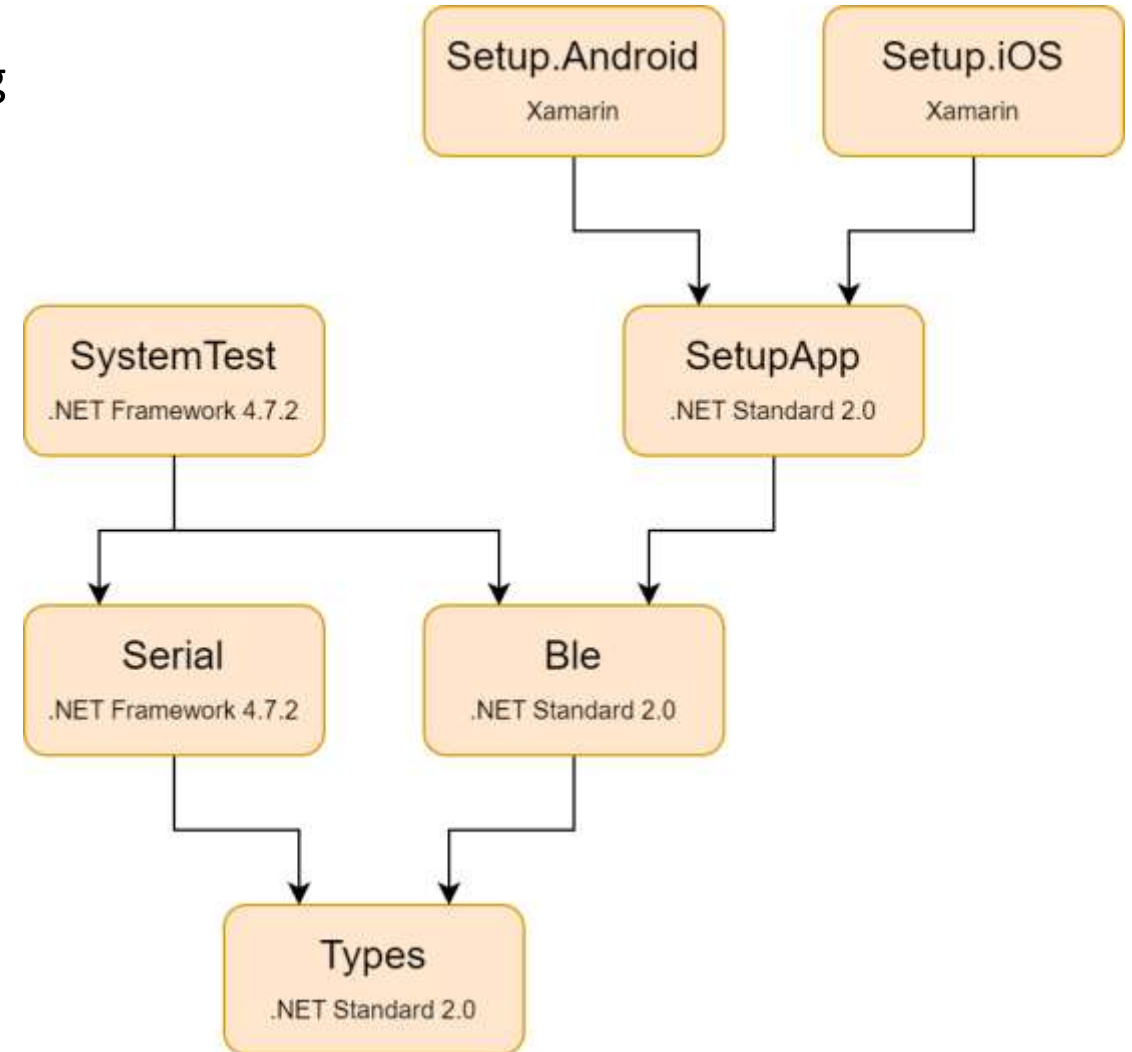


## Hardware Aufbau

- Total 54 Nordic nRF52 Development kits
    - 2 Gateway
    - 2 Präsenzerkennungen
    - 36 Aktoren Typ A
    - 10 Aktoren Typ B
    - 1 Sensor
    - 2 «Unbekannte Zukunftsmodule»
    - 1 Bluetooth LE Central (SmartPhone-Ersatz)
  - Ermöglicht maximale Installation mit 50 Modulen oder Modul mit 16 Komponenten
  - Prüfung Fehlerbehandlung von zweitem Gateway resp. zweiter Präsenzerkennung
- Ein laufendes Test-System mit gutem Kosten-Nutzen-Verhältnis

## Features

- Visual Studio Projekte für Libraries, Apps und Testing
- SetupApp für Android und iOS
  - Für manuelle Tests und Demonstrationen
  - Auf gleicher Code-Basis
- SystemTest Projekt in C# mit NUnit 3
  - Skalierbarkeit von 1 Gerät bis zu maximaler Installationsgrösse
    - Am Entwicklerarbeitsplatz einsetzbar



## Features

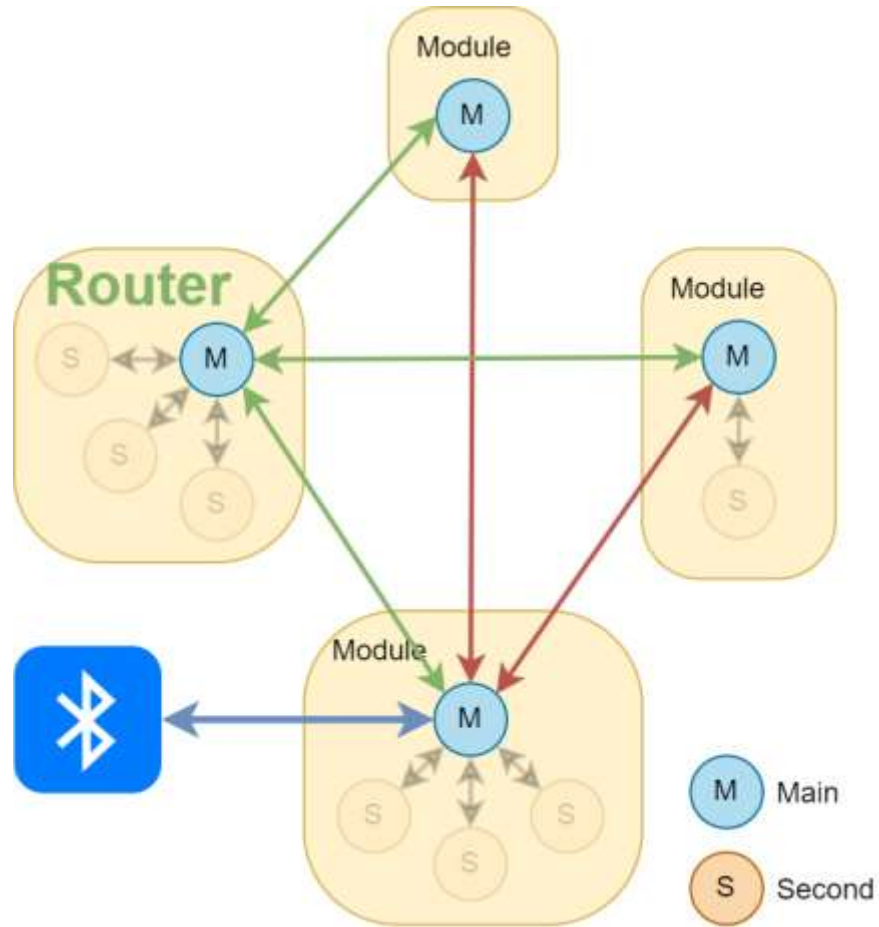
■ ApplicationLayer	89 Test Cases
■ ApplicationLayer.DataAccess	4 Test Cases
■ ApplicationLayer.DiagnosticData	49 Test Cases
■ ApplicationLayer.FirmwareUpdate	230 Test Cases
■ ApplicationLayer.Module	81 Test Cases
■ ApplicationLayer.Router	44 Test Cases
■ ApplicationLayer.Setup	183 Test Cases
■ LowerLayer	16 Test Cases
■ LowerLayer.Database	150 Test Cases
■ LowerLayer.Rf	328 Test Cases
■ TestFramework	28 Test Cases

**~1200 Test Cases**

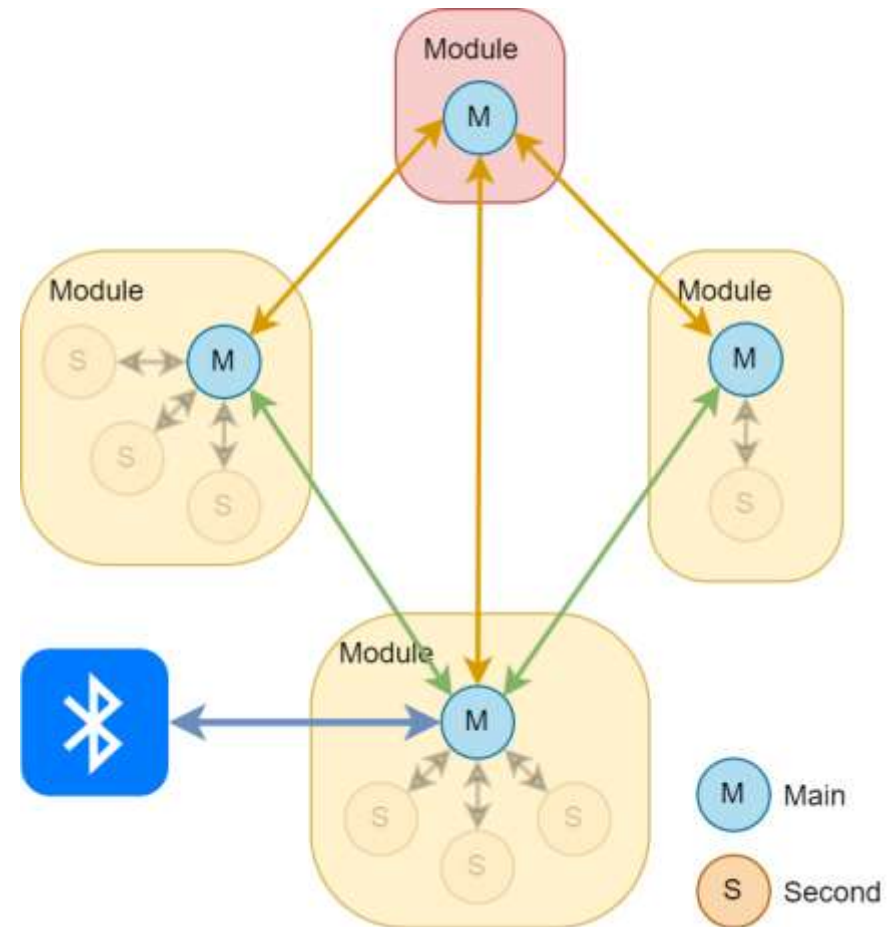
## Features

- Helper Funktionen
  - Dynamische Device-Collection
  - Bluetooth Aktivierung, Verbindungsaufbau & Initialisierung
  - Generische Implementation wiederholender Abläufe
  - ...
- Triggern auf Statusübergänge eines RTOS-Prozesses
- Triggern auf Empfang einer Funknachricht
- Erzeugung und Injektion einer vollständigen Installation innert weniger Sekunden
- Überwachung und Auswertung des Ressourcenverbrauchs auf den Endgeräten

## Features



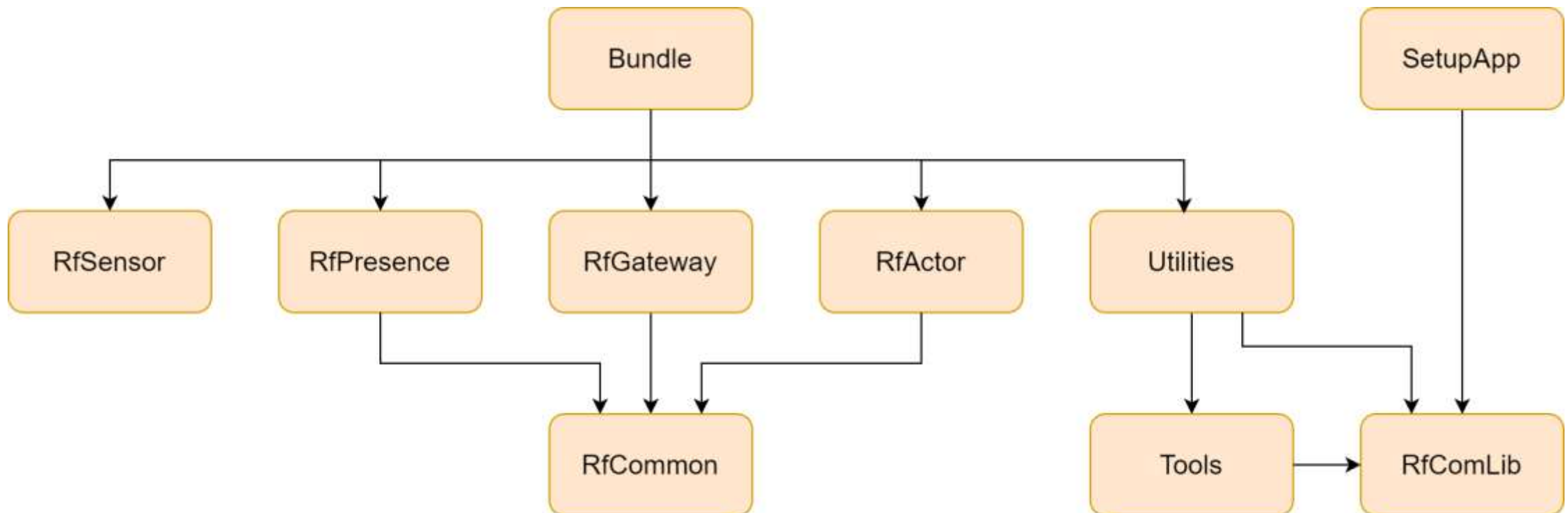
Einzelne RF-Strecken stumm schalten



RF von Komponenten stumm schalten

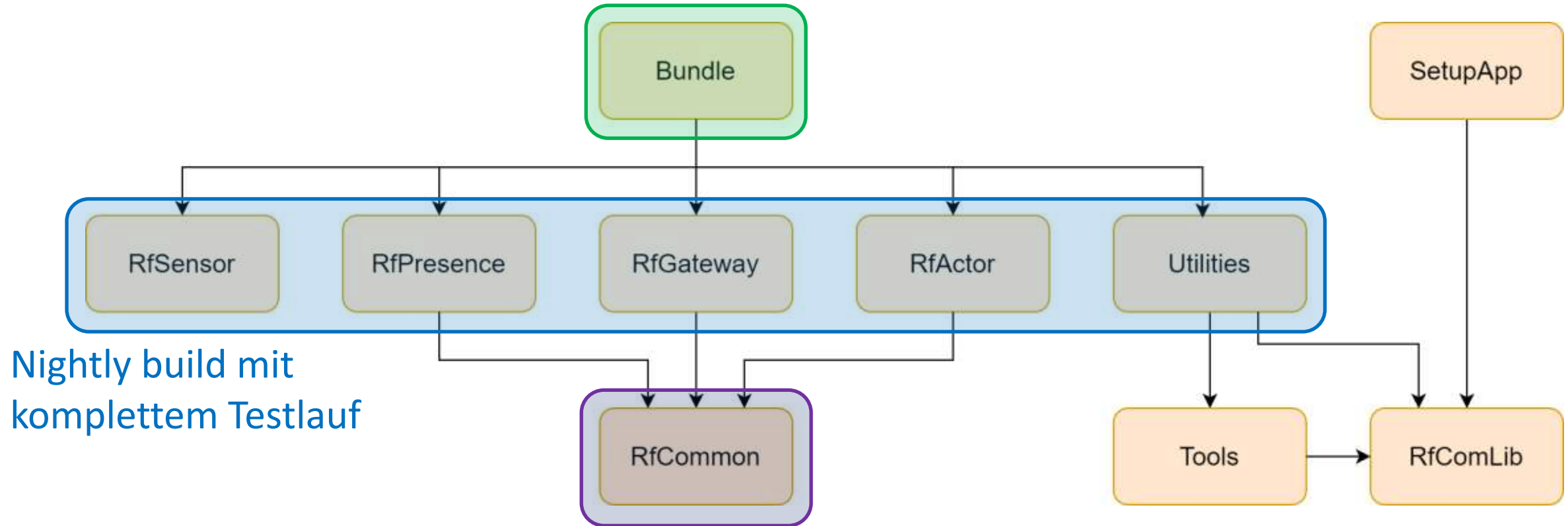
## Repositories & Submodules

- Versionierung als Software-Paket über Bundle mit Submodulen zu den verschiedenen Komponenten
- Komponenten nutzen generischen RfCommon als Submodul
- Utilities (Testing) und SetupApp nutzen RfComLib als Submodul
- RfSensor als einzige batteriebetriebene Komponente hat einen reduzierten Funktionsumfang



## Jenkins Pipelines

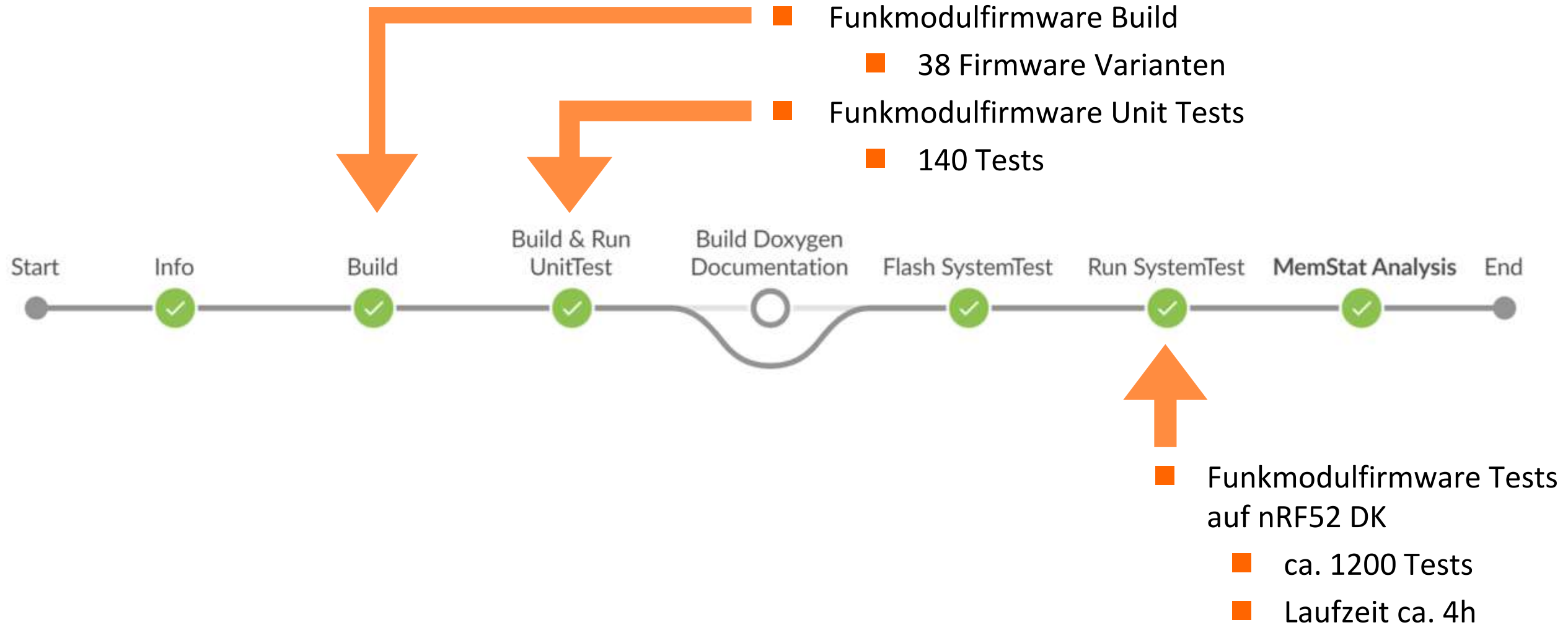
Build mit komplettem Testlauf on Commit



Nightly build mit komplettem Testlauf

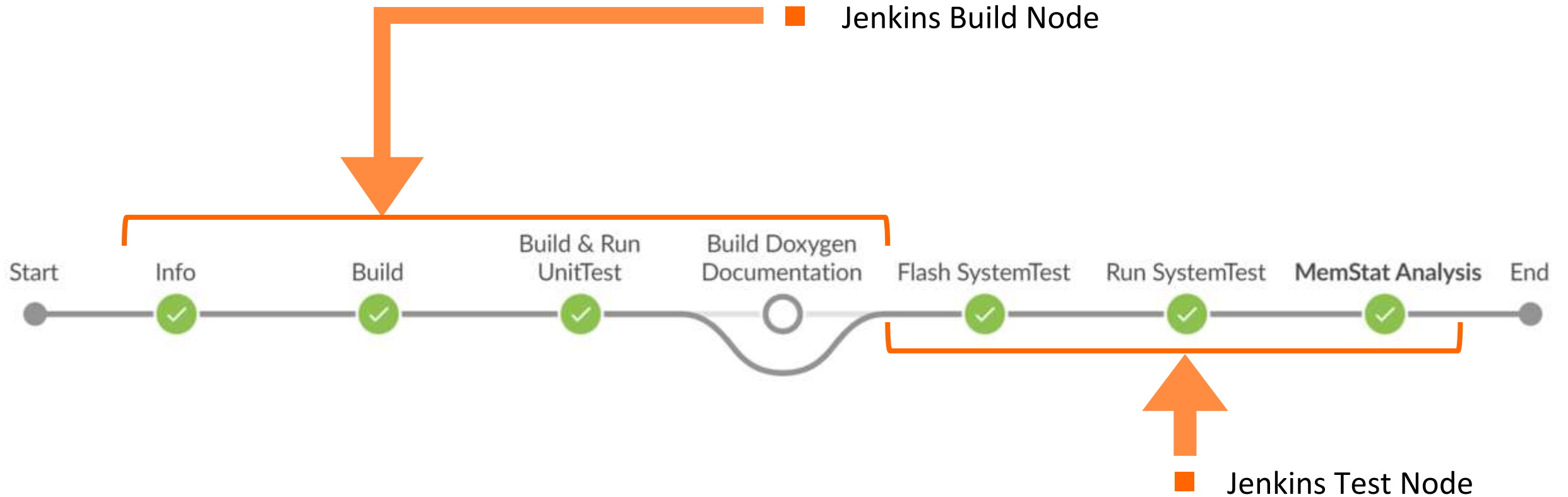
Build mit UnitTest-Durchlauf on Commit

## Jenkins Pipelines





## Jenkins Pipelines



## Kosten-Nutzen

- Ist eine «Muss» Anforderung: Komplexe vernetzte Systeme sind konventionell nicht mehr testbar
- Beträchtliche Investitionskosten, zahlt sich über Projektlaufzeit aber aus
- Für konkrete Zahlen am Arendi Stand rein schauen....



**We are your solution.**

**Arendi AG**  
Eichtalstrasse 55  
8634 Hombrechtikon  
Switzerland

[www.arendi.ch](http://www.arendi.ch)

r&d