

SDN

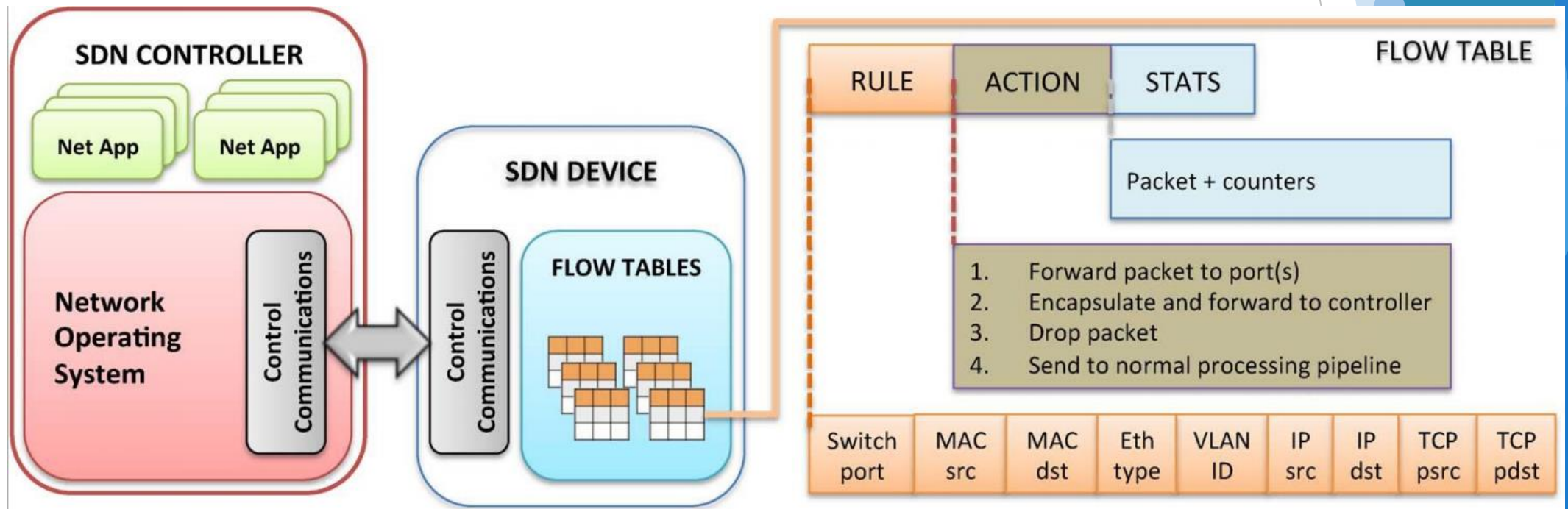
Überblick, Ziele und Meilensteine

Gliederung

- ▶ Überblick
- ▶ Erste Erfahrungen
- ▶ Ziele
- ▶ Meilensteine / Schedule

Zur Erinnerung

- OpenFlow definiert die Schnittstelle zwischen Controller und Forwarding Device



Überblick

- ▶ SDN Controller nutzt OpenFlow Protokoll
- ▶ Mindestens 16 mögliche Softwarelösungen (Wikipedia)
- ▶ OpenDaylight ist Basis für viele Lösungen (Java)
- ▶ Ryu als Python Alternative
- ▶ Weitere mögliche Lösungen nur oberflächlich angeschaut

OpenDaylight

▶ Java-basiert

- ▶ Net App muss auch in Java geschrieben werden
- ▶ Java Interfaces etc. nutzen

▶ Maven als Build-System (Deployment)

▶ YANG als Markup-Language für Hardwarebeschreibung

- ▶ Nodes (Switches)
- ▶ Node-connectors (Switch-Ports)

▶ Ergebnis:

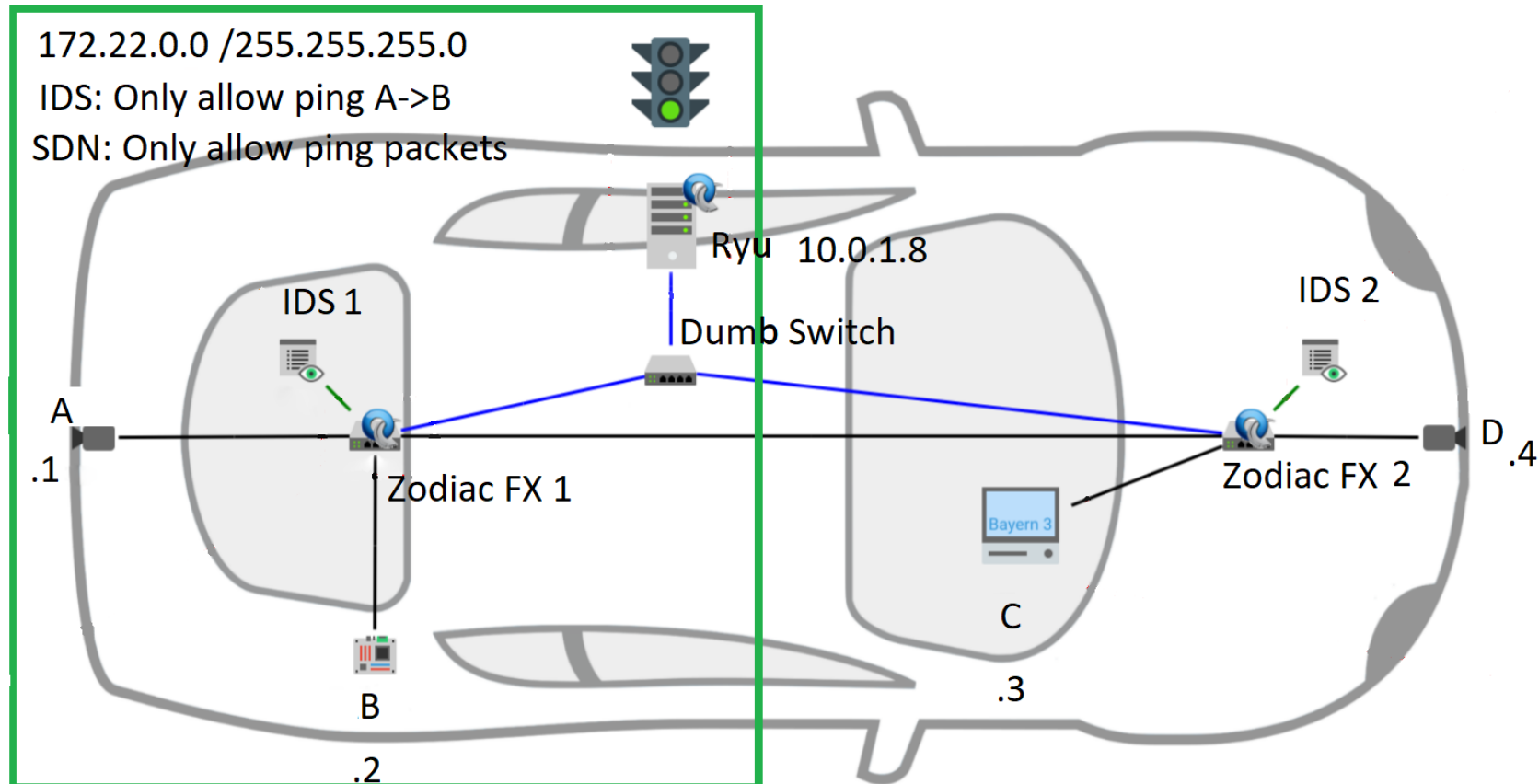
- ▶ Überladen
- ▶ Kompliziertes Deployment
- ▶ Crasht in ersten Tests auf Raspberry Pi

Ryu

- ▶ Python-basiert
 - ▶ Net App muss auch in Python geschrieben werden
- ▶ Ressourcenschonend
- ▶ Liegt in Debian Repositories vor
- ▶ Topologieansicht kann generiert werden
- ▶ Einfaches Deployment
 - ▶ Schreiben eines Python-Scriptes
- ▶ Ergebnis: Wird als Lösung anvisiert

Ziele

► Topologie / Architektur nachbilden



Meilensteine / Schedule

KW	Date	Details
14	01.04.2019	Kommunikationskonzept (Komm. Partner, Test-Pakete)
15	08.04.2019	Schulung TrustNode TSSDN Switch
16	15.04.2019	Implementierung Kommunikationskonzept
17	22.04.2019	Ostermontag (Frei)
18	29.04.2019	Testkonzept / Demokonzept
19	06.05.2019	Meilenstein 2: Erste mock-up demo
20	13.05.2019	Bug Fixes / Feature Erweiterungen
21	20.05.2019	Bug Fixes / Feature Erweiterungen
22	27.05.2019	Bug Fixes
23	03.06.2019	Meilenstein 3: Release Candidate 1
24	11.06.2019	Bug Fixes / Präsentationskonzept
25	17.06.2019	Präsentation üben / testen
26	24.06.2019	Finale Projektpräsentation

Teamaufteilung

- ▶ Testdaten: Martin, Randolph
- ▶ Architektur / Regeln : Simon, Magnus
- ▶ Ryu / Net App : Kai, Magnus

Gitlab

- ▶ <https://gitlab.informatik.haw-hamburg.de/acf016/po-caz-sdn>