Proteus: A Network Service Control Platform for Service Evolution in a Mobile Software Defined Infrastructure

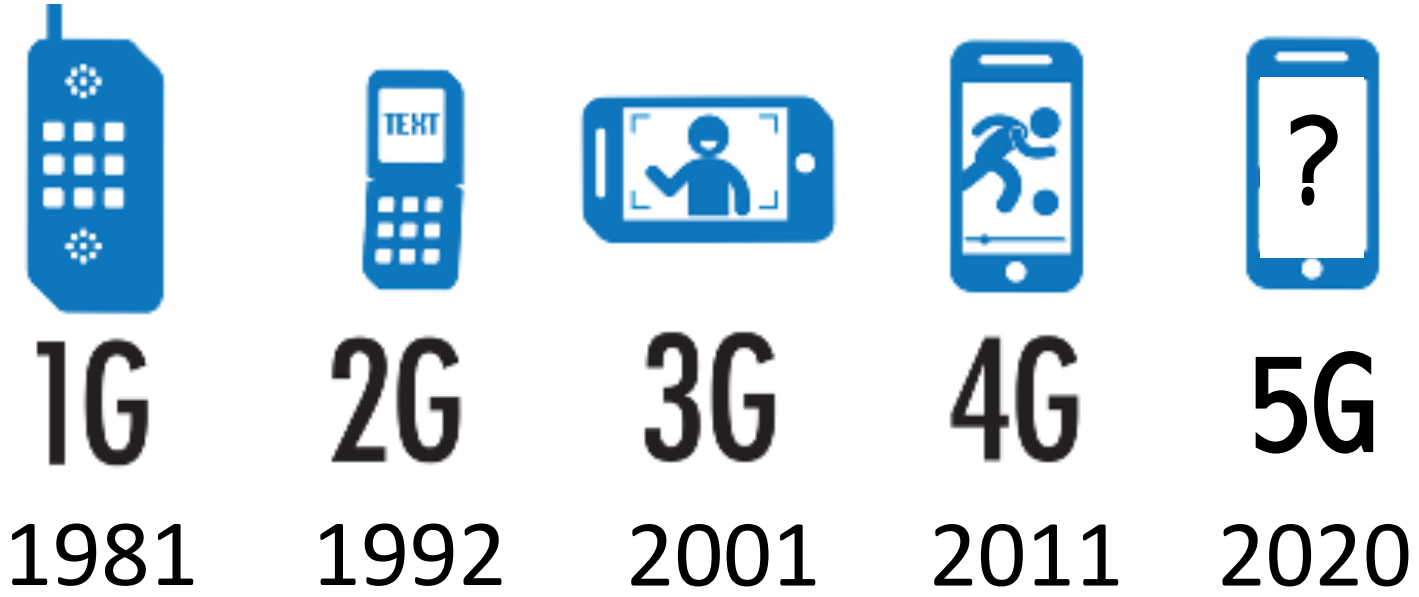
Aisha Syed and Kobus Van der Merwe

THE UNIVERSITY OF UTAH

This work was supported by the National Science Foundation under grant numbers 1305384 and 1302688.

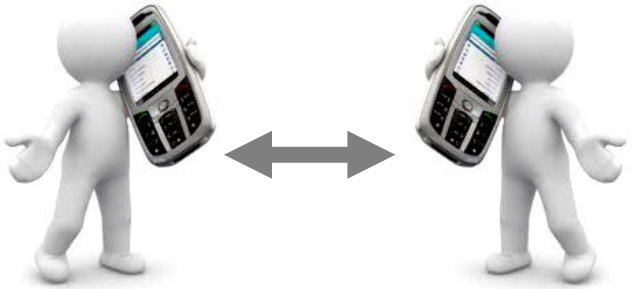
Motivation

- Mobile network service deployment and evolution very slow



Motivation

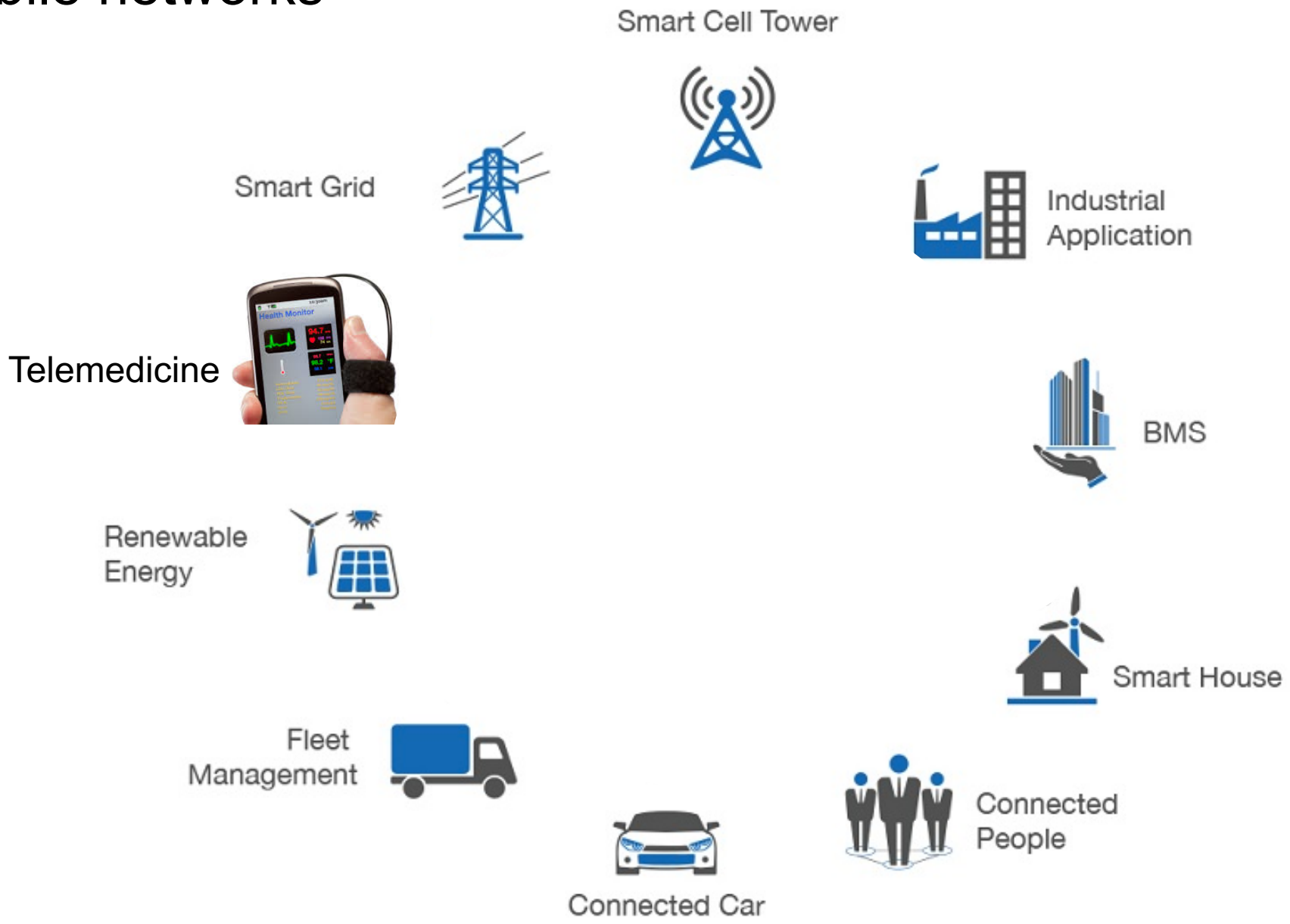
- Monolithic service abstraction: mobile broadband



Human-to-human

Human-to-machine

- Future mobile networks



- Future mobile networks

- Future mobile networks

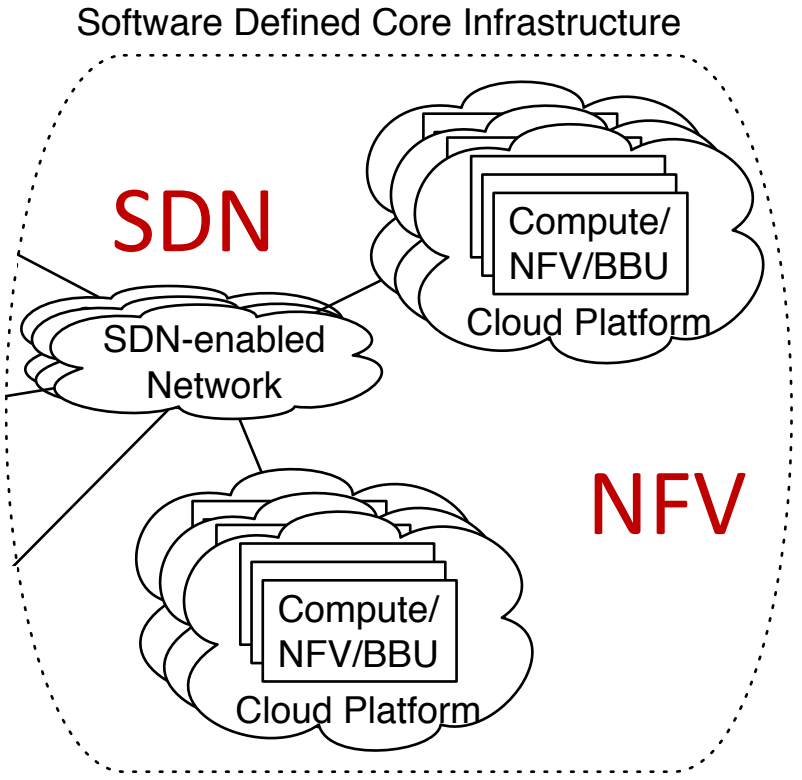
- Future mobile networks

Motivation

- With SDN and in-network clouds enabling NFV, we realize **trend towards a mobile software-defined infrastructure (SDI):**

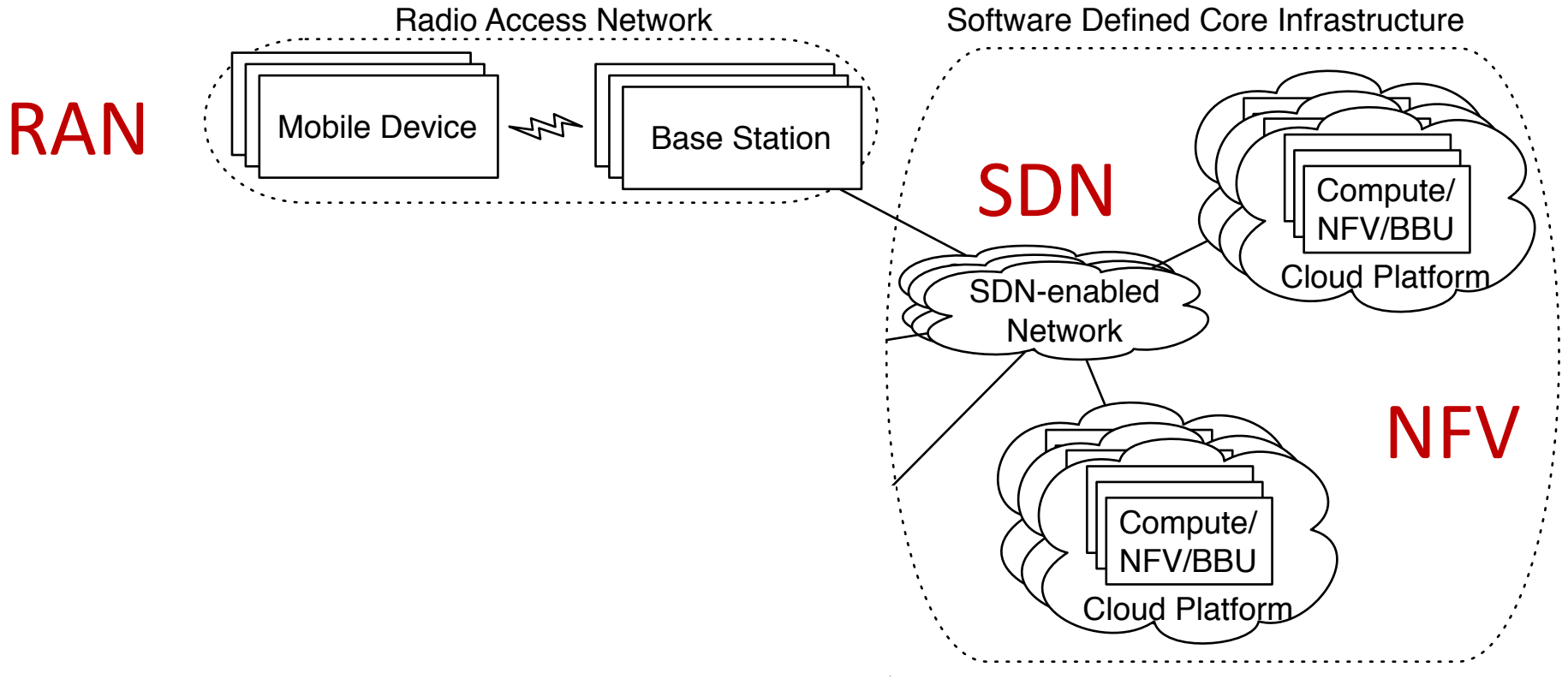
Motivation

- With SDN and in-network clouds enabling NFV, we realize **trend towards a mobile software-defined infrastructure (SDI):**



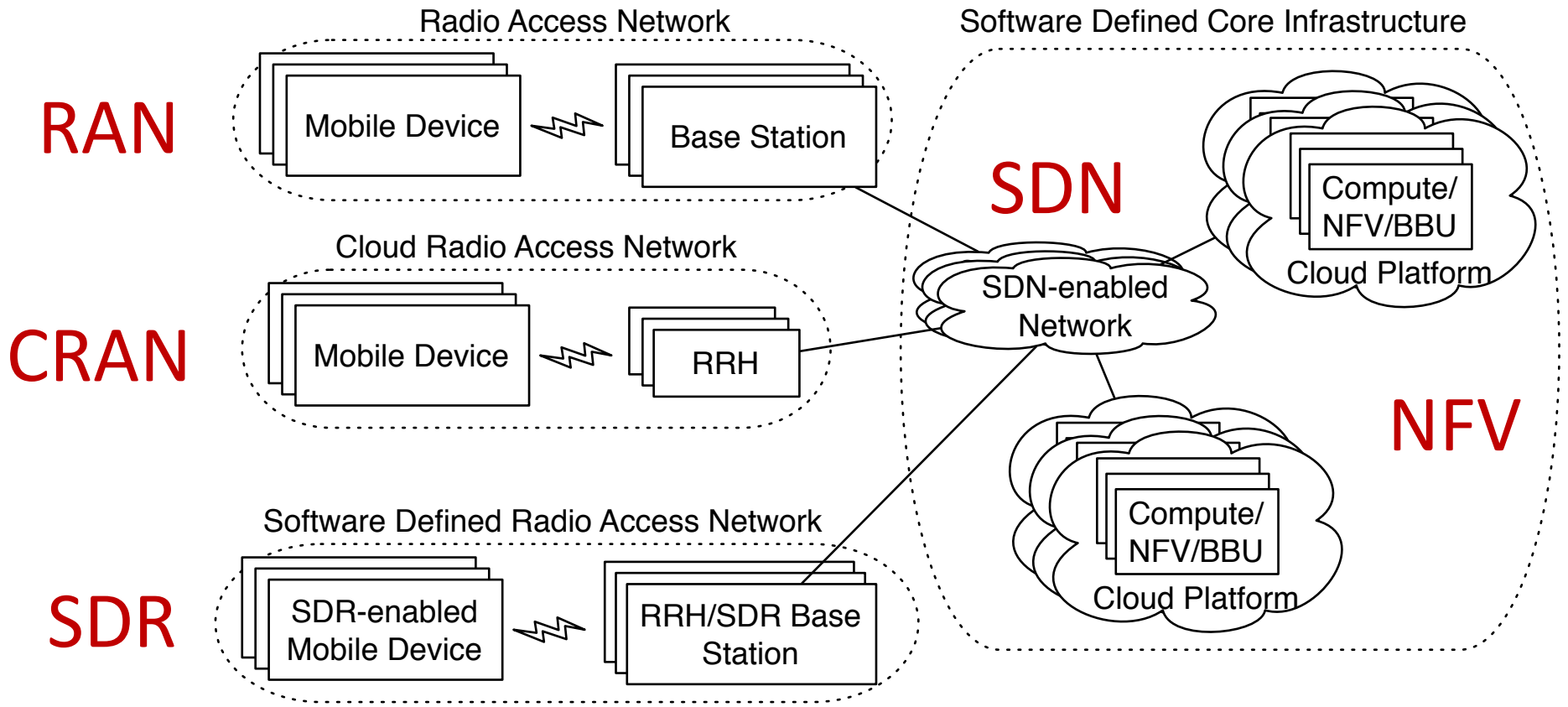
Motivation

- With SDN and in-network clouds enabling NFV, we realize **trend towards a mobile software-defined infrastructure (SDI):**



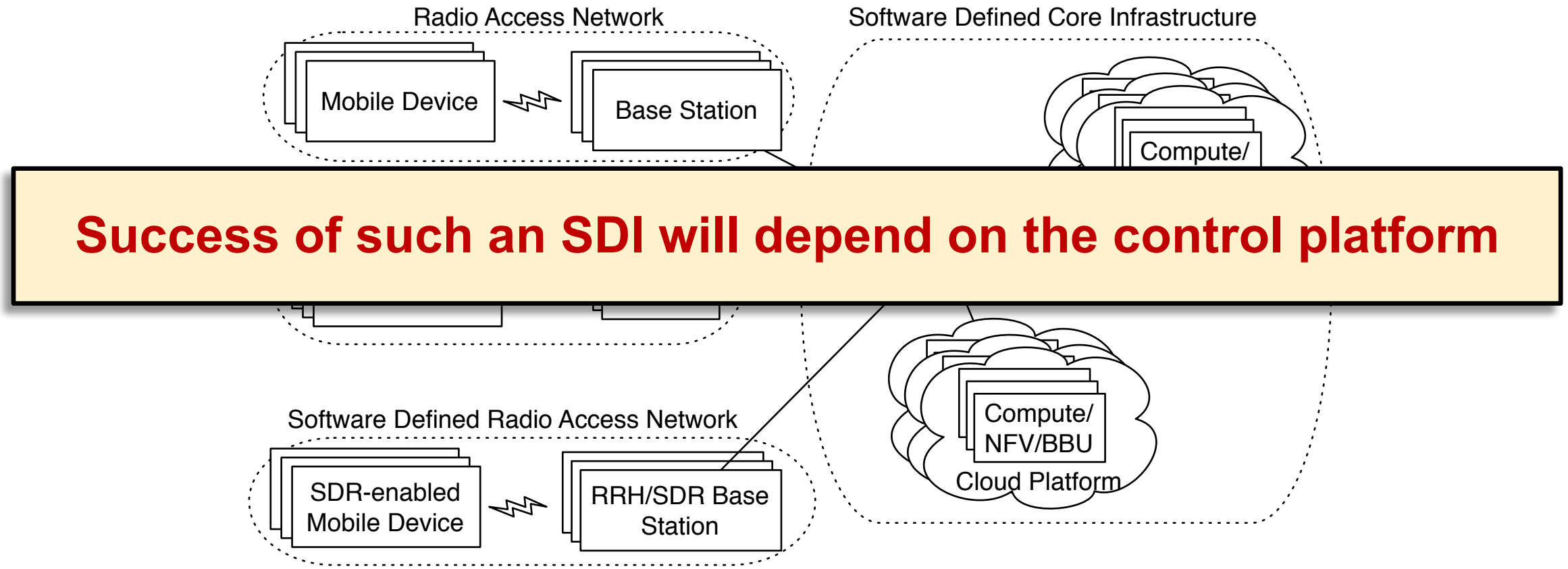
Motivation

- With SDN and in-network clouds enabling NFV, we realize **trend towards a mobile software-defined infrastructure (SDI):**



Motivation

- With SDN and in-network clouds enabling NFV, we realize **trend towards a mobile software-defined infrastructure (SDI):**

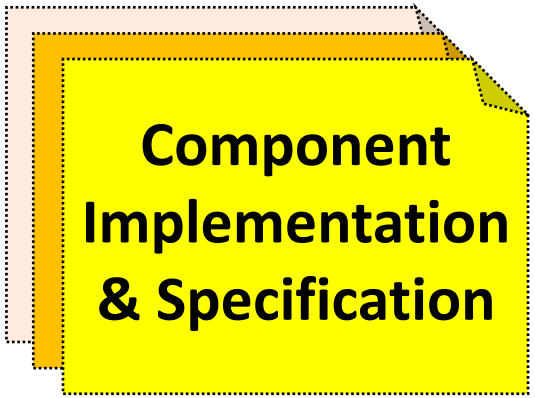


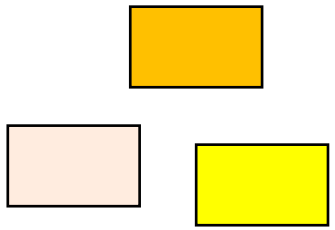
Proteus

- A mobile network service control platform to enable safe and rapid service creation and evolution in a mobile SDI

Proteus

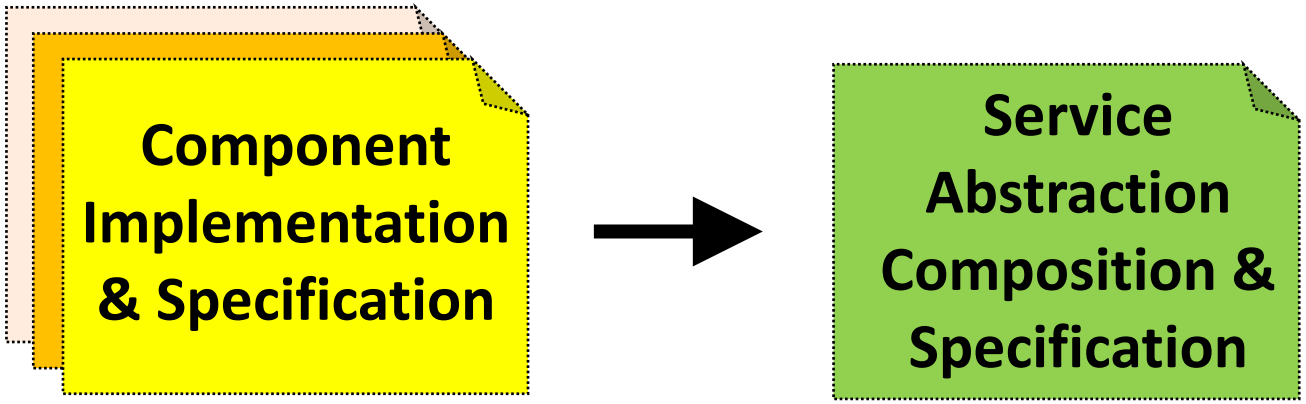
- A mobile network service control platform to enable safe and rapid service creation and evolution in a mobile SDI

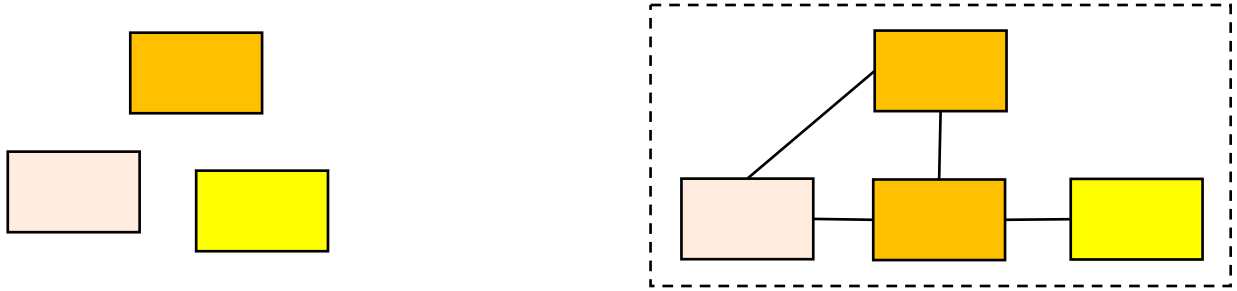




Proteus

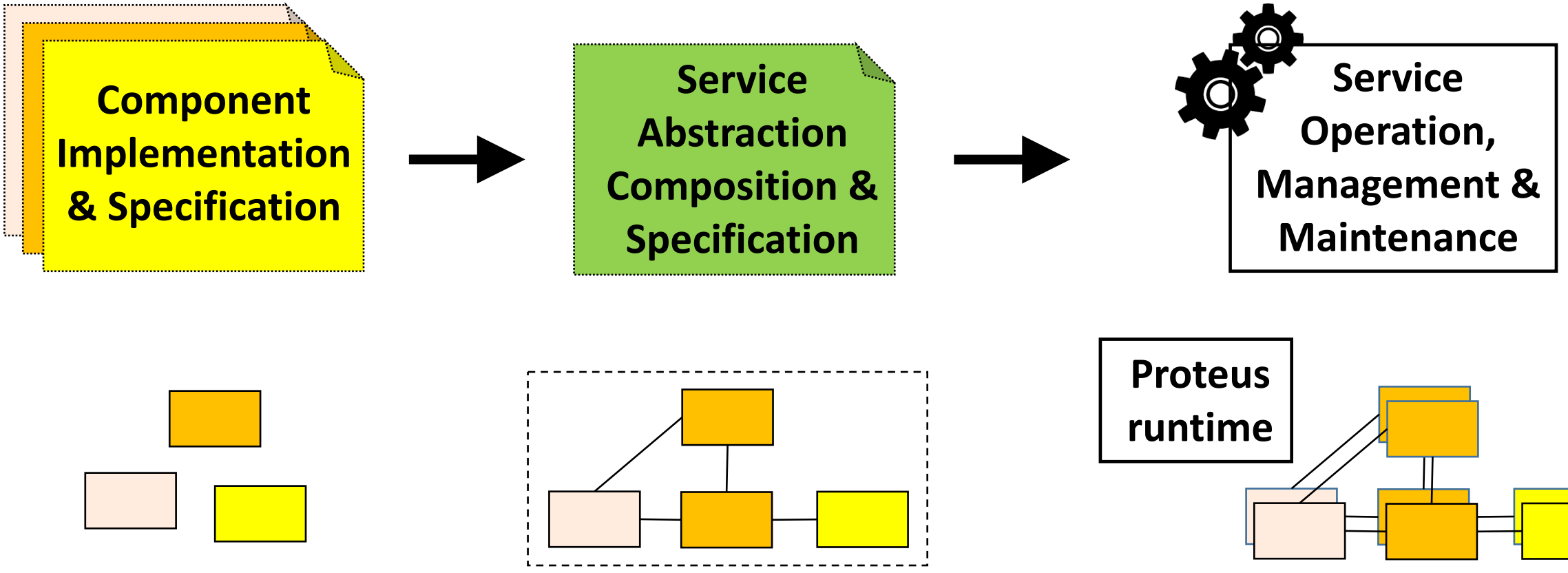
- A mobile network service control platform to enable safe and rapid service creation and evolution in a mobile SDI





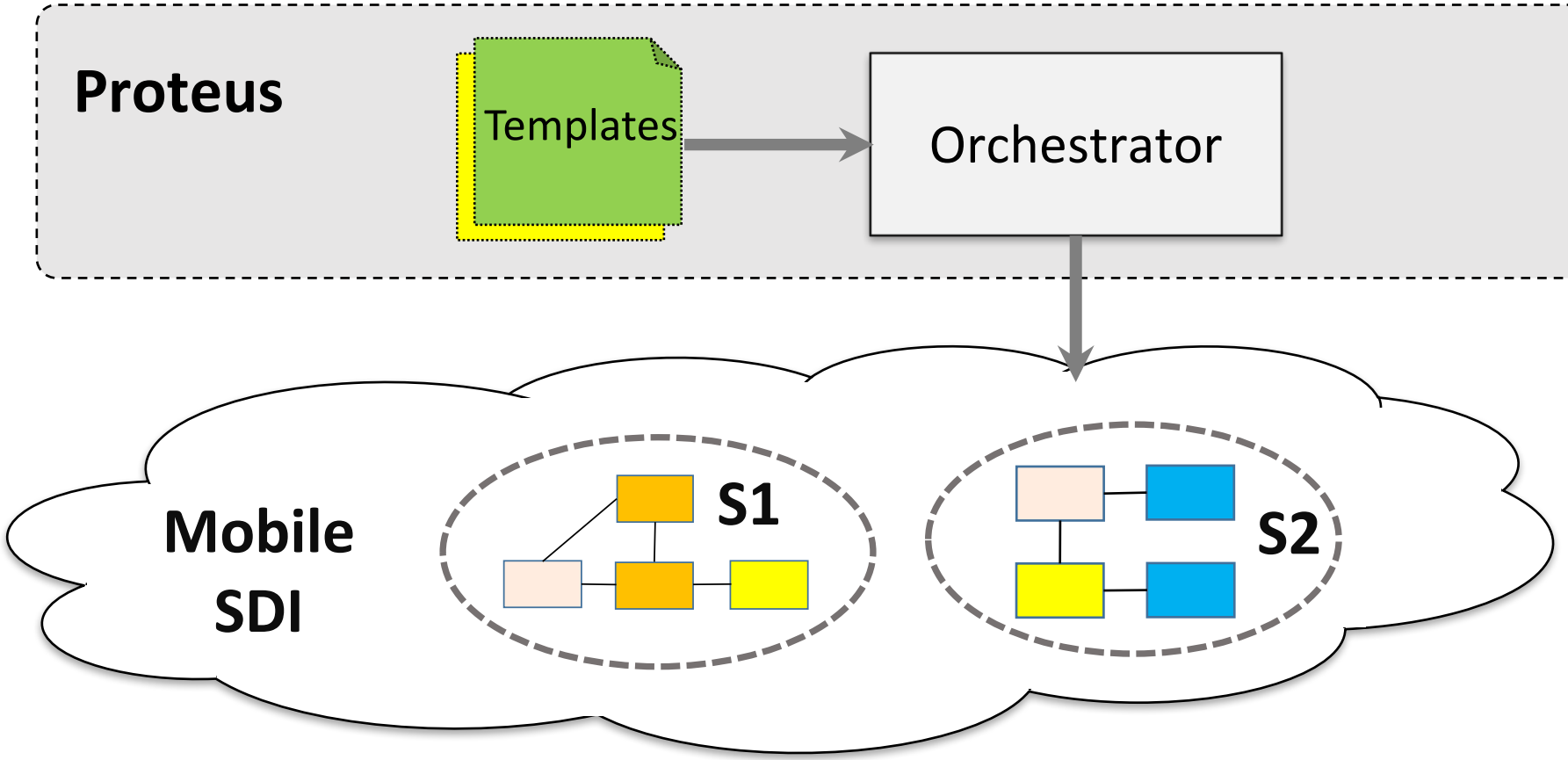
Proteus

- A mobile network service control platform to enable safe and rapid service creation and evolution in a mobile SDI



Proteus

- A mobile network service control platform to enable safe and rapid service creation and evolution in a mobile SDI

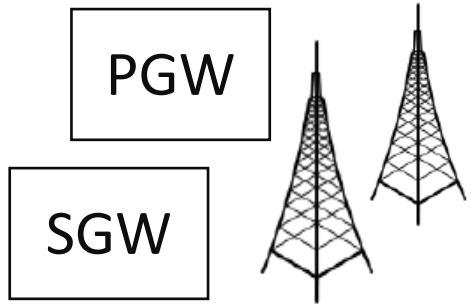


Challenges and Design Principles

- Clean separation between infrastructure and service abstractions

Challenges and Design Principles

- Clean separation between infrastructure and service abstractions



Physical Mobile resources

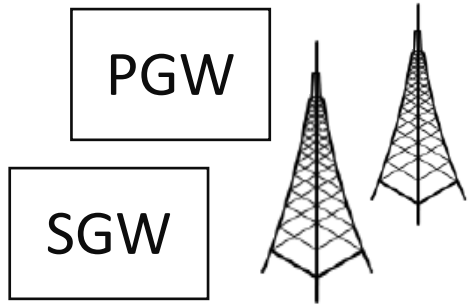
Compute

Network

Challenges and Design Principles

- Clean separation between infrastructure and service abstractions

Proteus Abstraction Layer



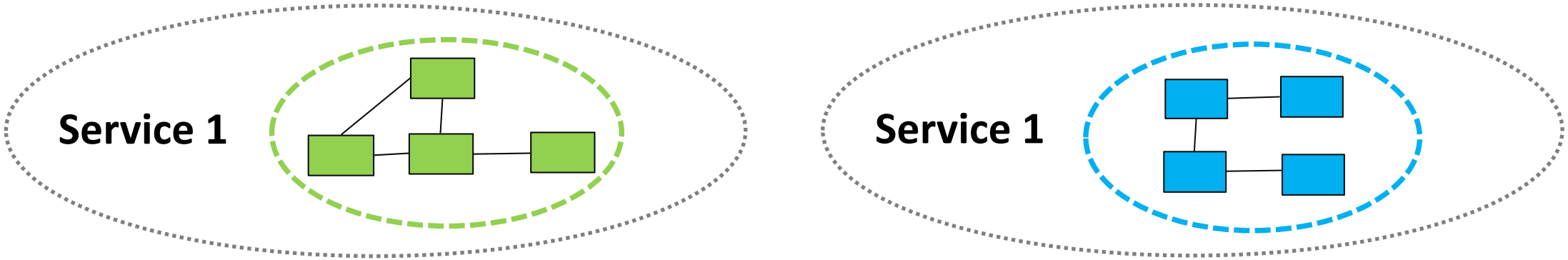
Physical Mobile resources

Compute

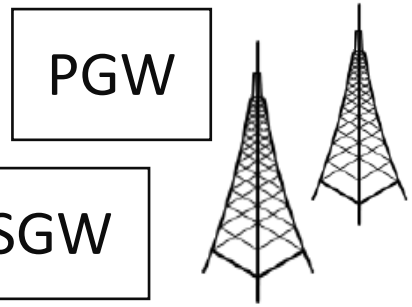
Network

Challenges and Design Principles

- Clean separation between infrastructure and service abstractions



Proteus Abstraction Layer



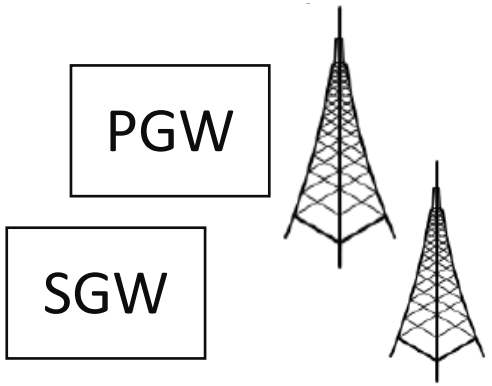
Physical Mobile resources

Compute

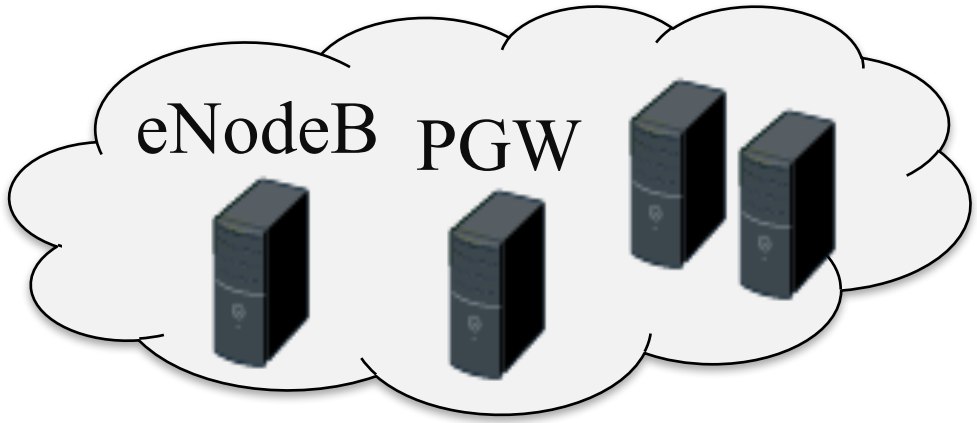
Network

Challenges and Design Principles

- Diversity in implementation and requirements



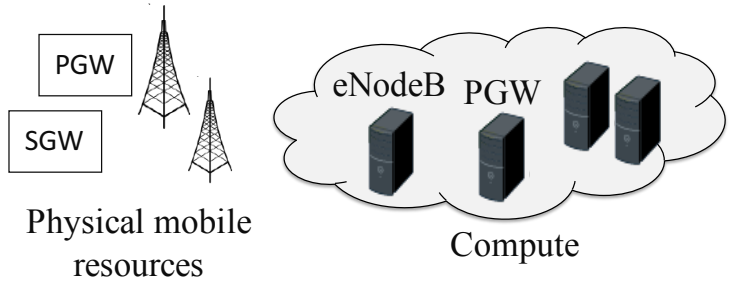
Physical mobile resources



Compute

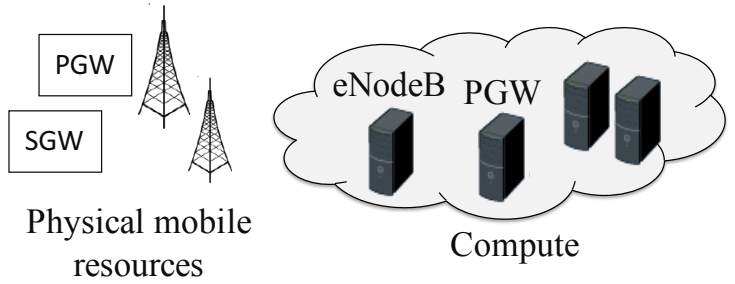
Challenges and Design Principles

- Diversity in implementation and requirements



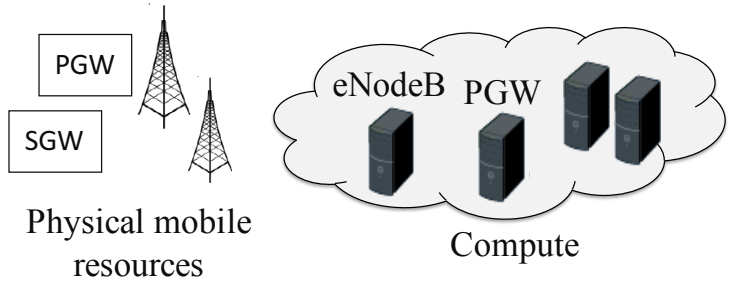
Challenges and Design Principles

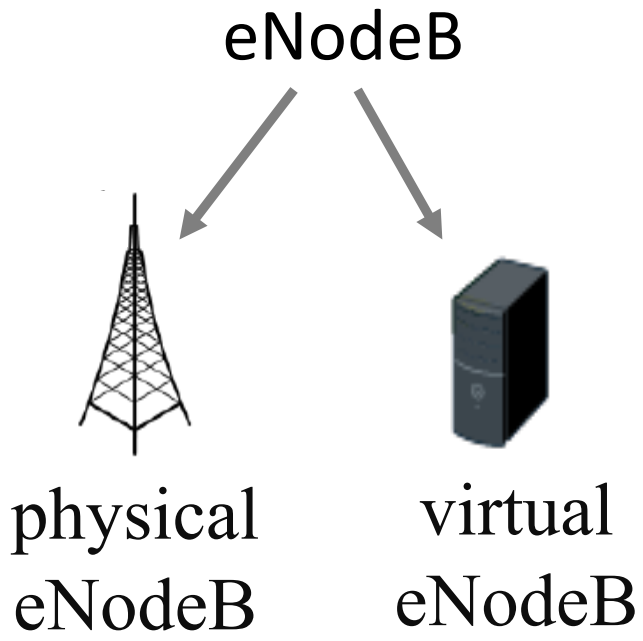
- Diversity in implementation and requirements
  - Polymorphic templates



Challenges and Design Principles

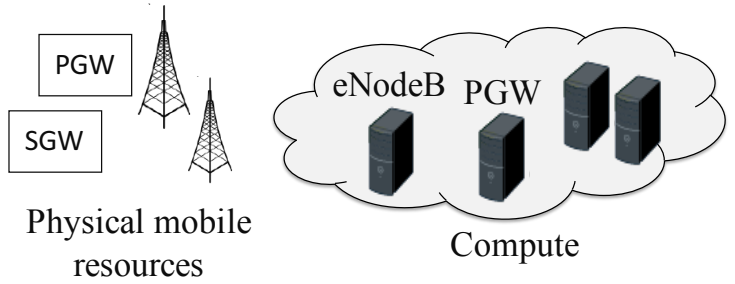
- Diversity in implementation and requirements
  - Polymorphic templates

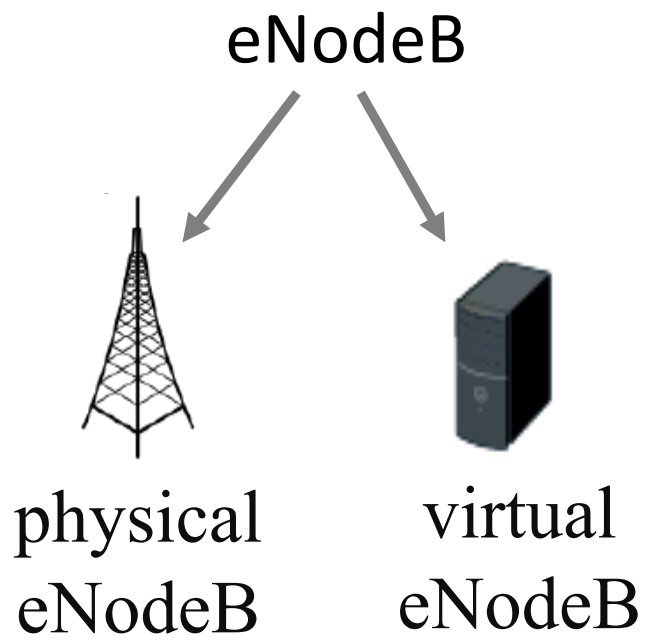


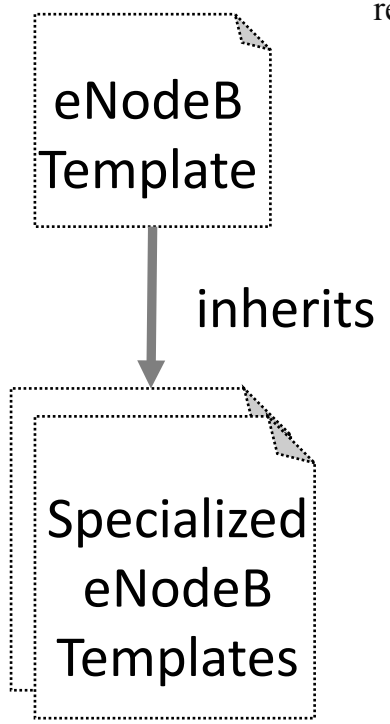


Challenges and Design Principles

- Diversity in implementation and requirements
  - Polymorphic templates





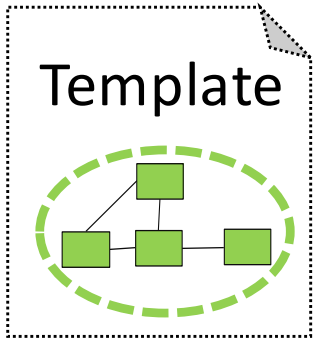


Challenges and Design Principles

- Mobility specific requirements

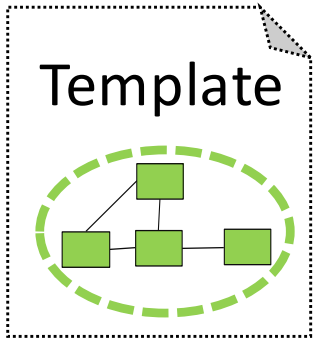
Challenges and Design Principles

- Mobility specific requirements
  - Templates capture logical service topology and dependencies



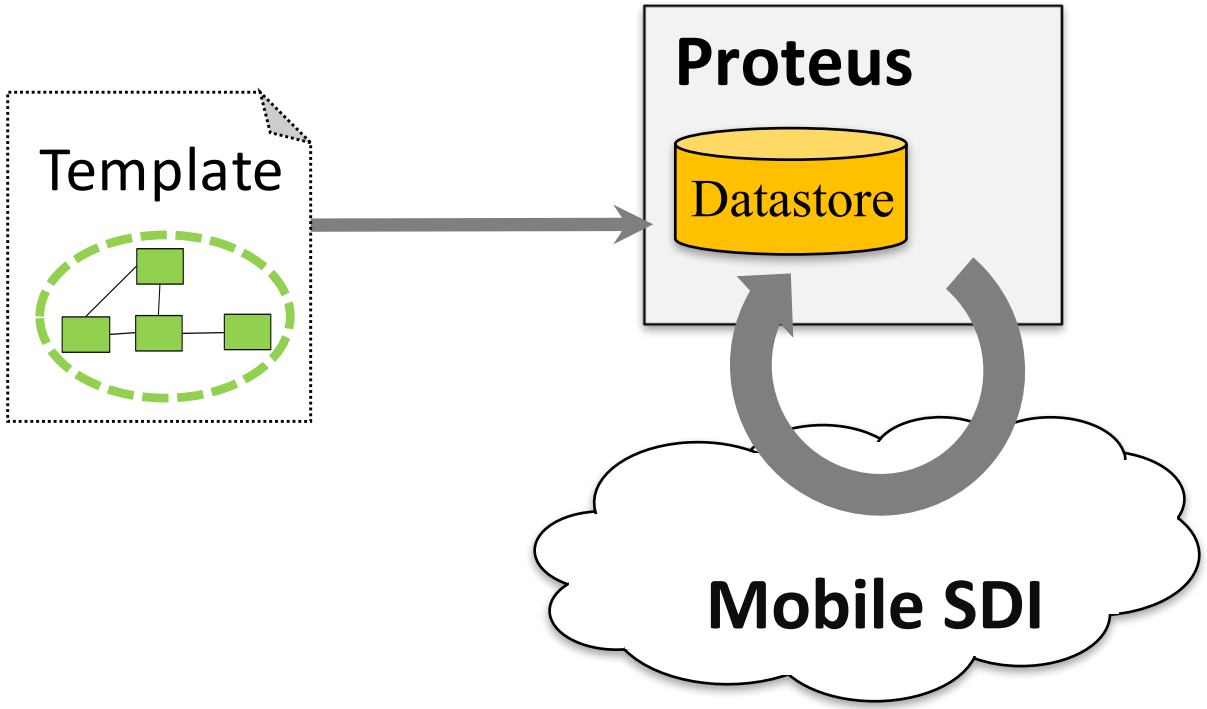
Challenges and Design Principles

- Mobility specific requirements
  - Templates capture logical service topology and dependencies
  - Templates and platform itself inherently data centric



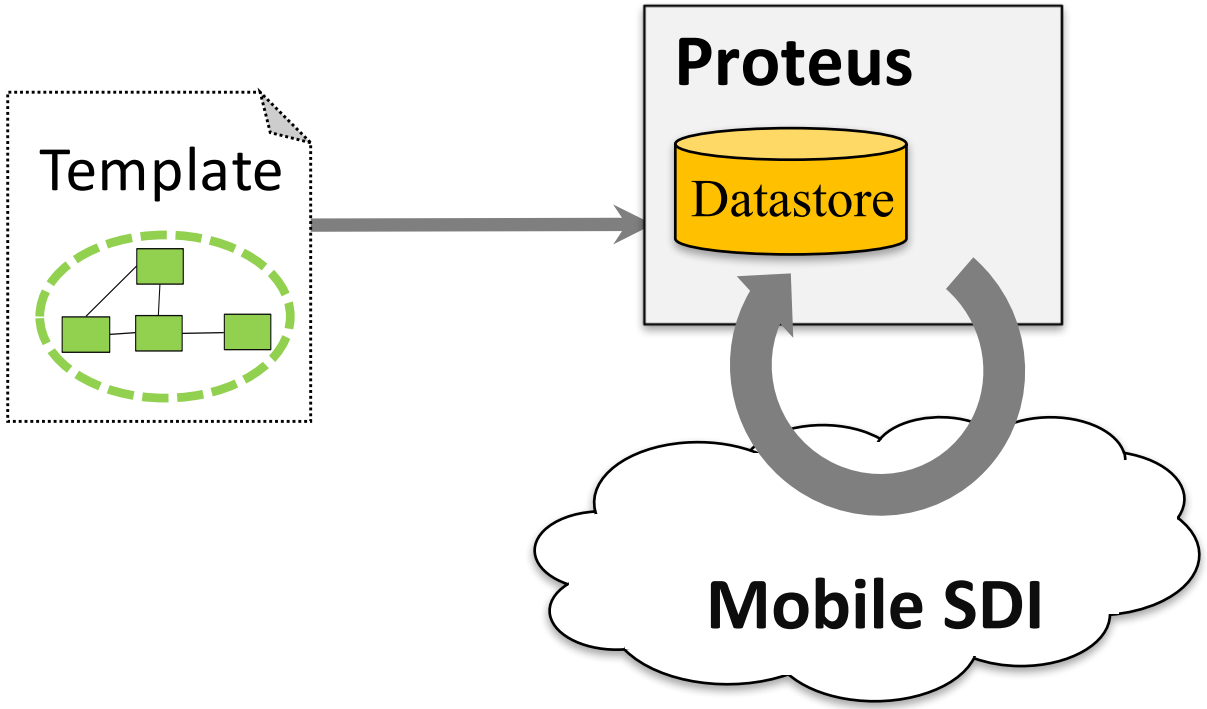
Challenges and Design Principles

- Mobility specific requirements
  - Templates capture logical service topology and dependencies
  - Templates and platform itself inherently data centric



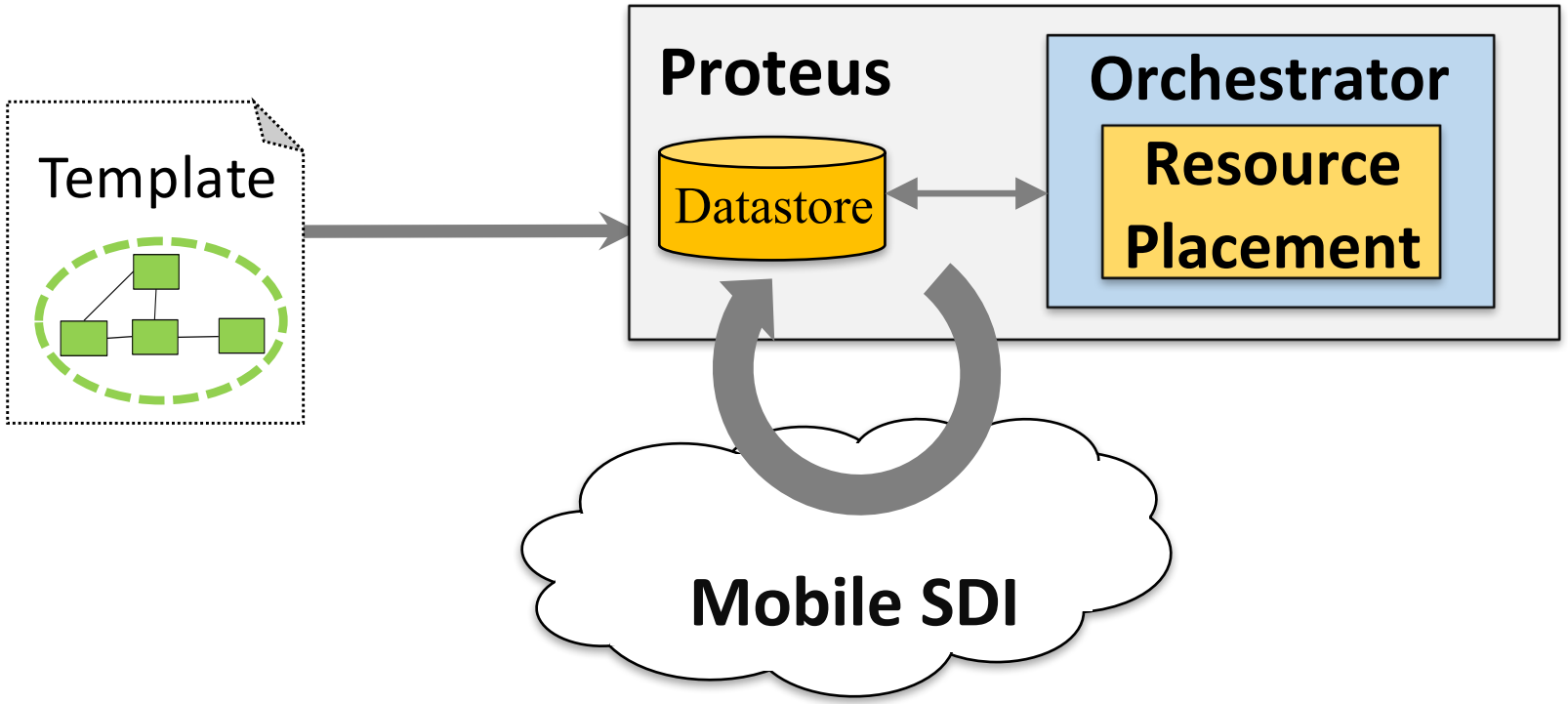
Challenges and Design Principles

- Mobility specific requirements
  - Templates capture logical service topology and dependencies
  - Templates and platform itself inherently data centric
  - Service agnostic resource placement



Challenges and Design Principles

- Mobility specific requirements
  - Templates capture logical service topology and dependencies
  - Templates and platform itself inherently data centric
  - Service agnostic resource placement

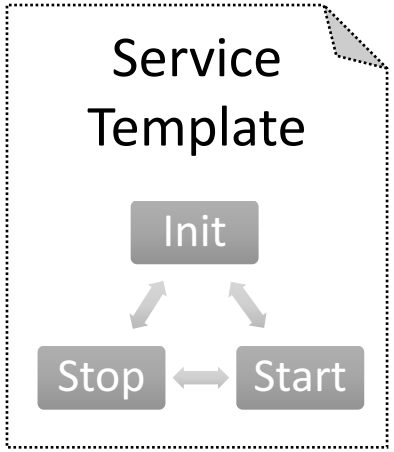


Challenges and Design Principles

- Safe service evolution and hosted multiplicity

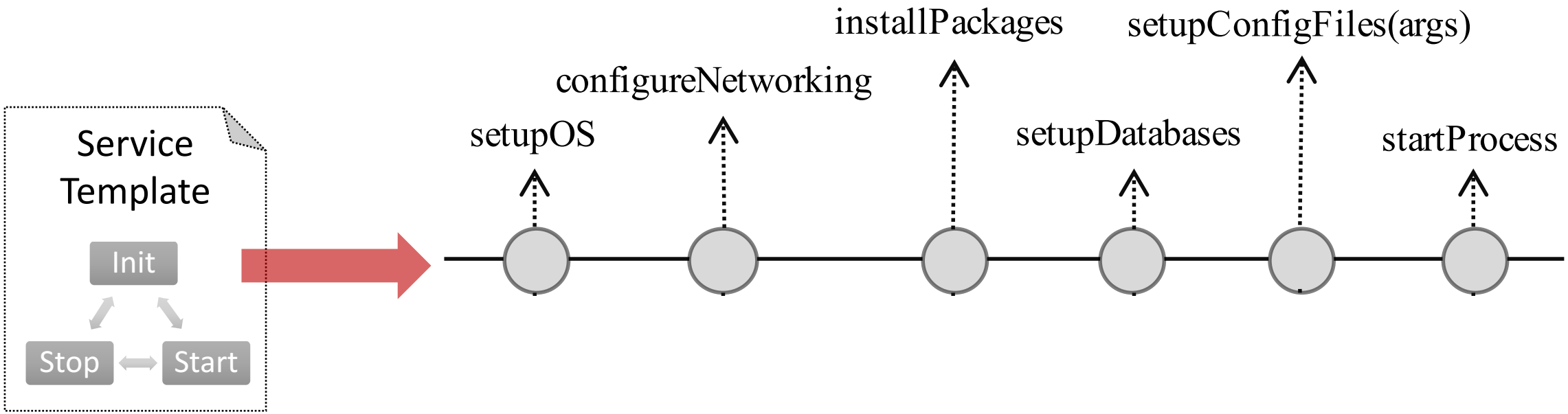
Challenges and Design Principles

- Safe service evolution and hosted multiplicity
  - Service realization plan



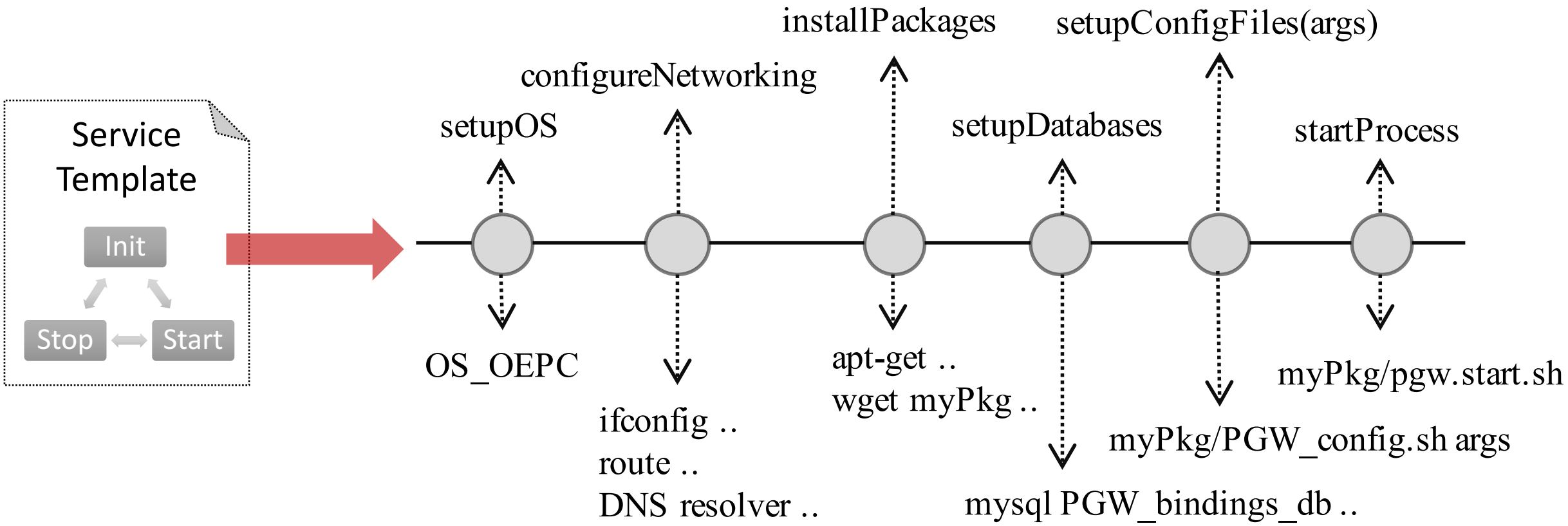
Challenges and Design Principles

- Safe service evolution and hosted multiplicity
  - Service realization plan



Challenges and Design Principles

- Safe service evolution and hosted multiplicity
  - Service realization plan

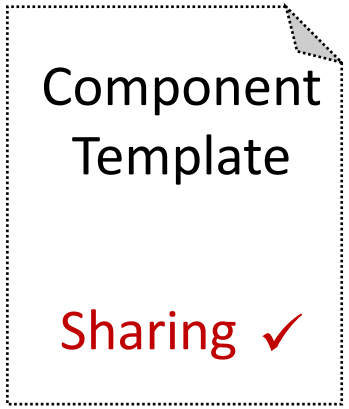


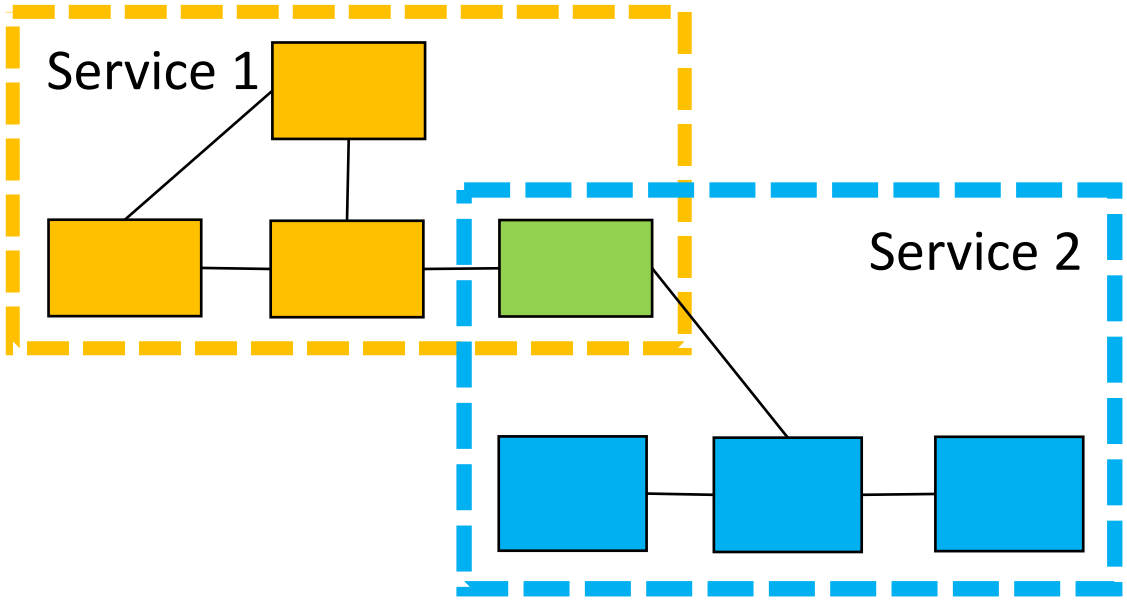
Challenges and Design Principles

- Safe service evolution and hosted multiplicity
  - Service realization plan
  - Component sharing

Challenges and Design Principles

- Safe service evolution and hosted multiplicity
  - Service realization plan
  - Component sharing



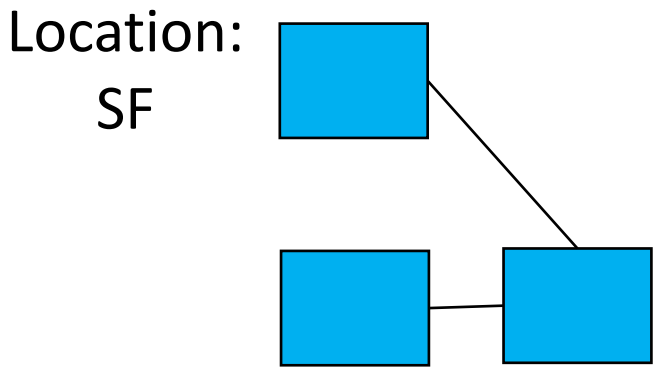


Challenges and Design Principles

- Safe service evolution and hosted multiplicity
  - Service realization plan
  - Component sharing
  - Component migration and traffic redirection primitives

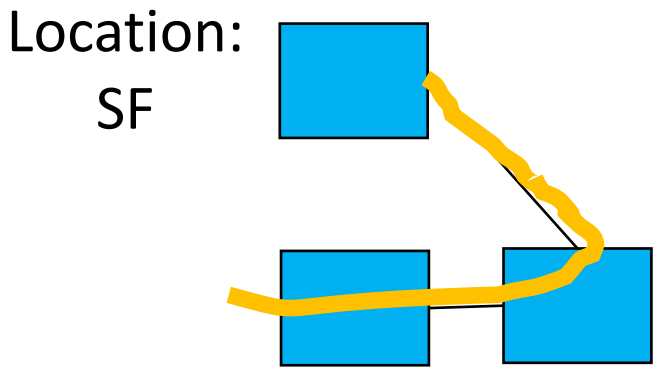
Challenges and Design Principles

- Safe service evolution and hosted multiplicity
  - Service realization plan
  - Component sharing
  - Component migration and traffic redirection primitives



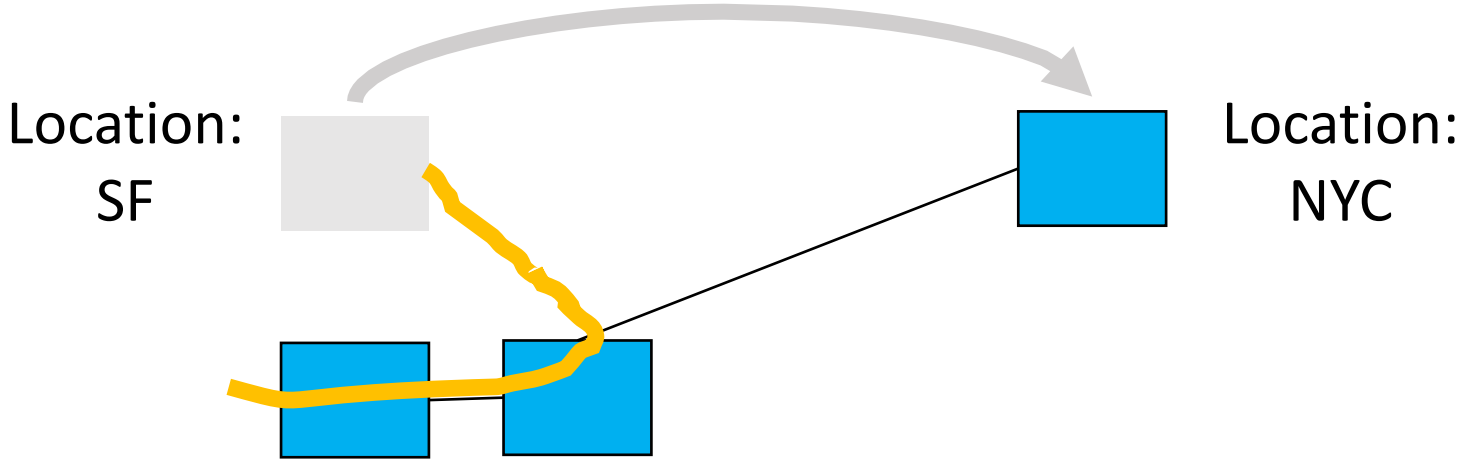
Challenges and Design Principles

- Safe service evolution and hosted multiplicity
  - Service realization plan
  - Component sharing
  - Component migration and traffic redirection primitives



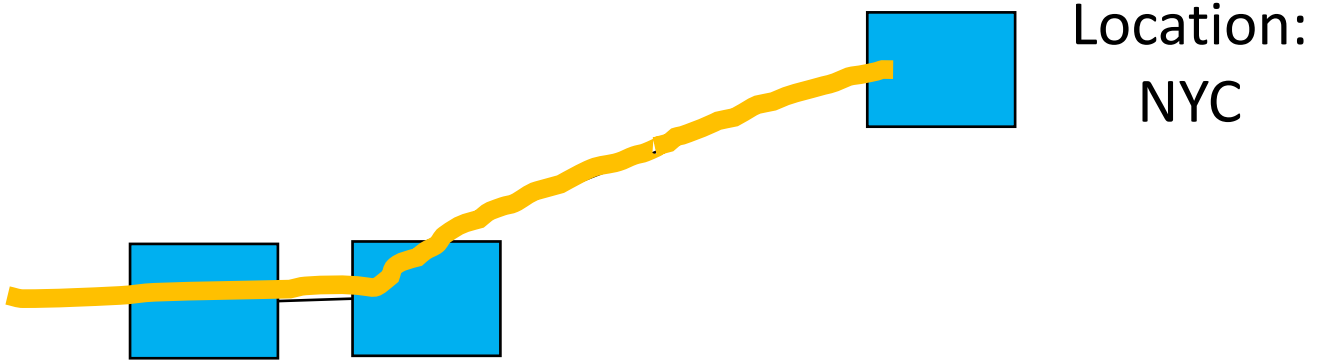
Challenges and Design Principles

- Safe service evolution and hosted multiplicity
  - Service realization plan
  - Component sharing
  - Component migration and traffic redirection primitives

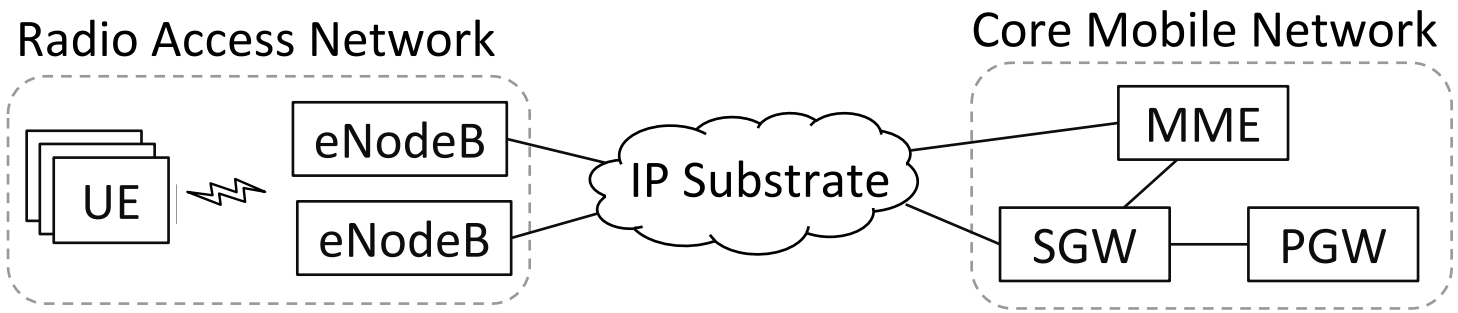


Challenges and Design Principles

- Safe service evolution and hosted multiplicity
  - Service realization plan
  - Component sharing
  - Component migration and traffic redirection primitives

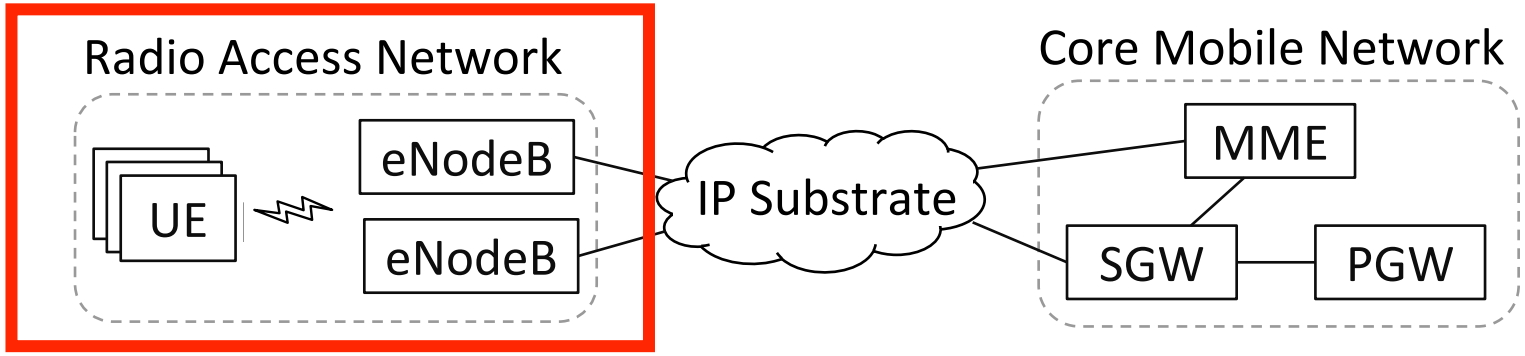


Use Cases



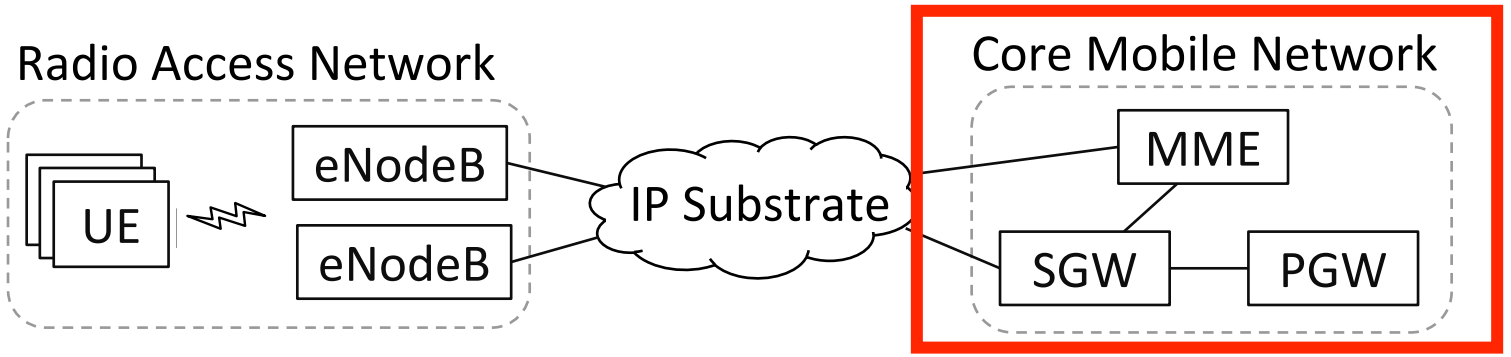
LTE/EPC: Standard mobile broadband service

Use Cases



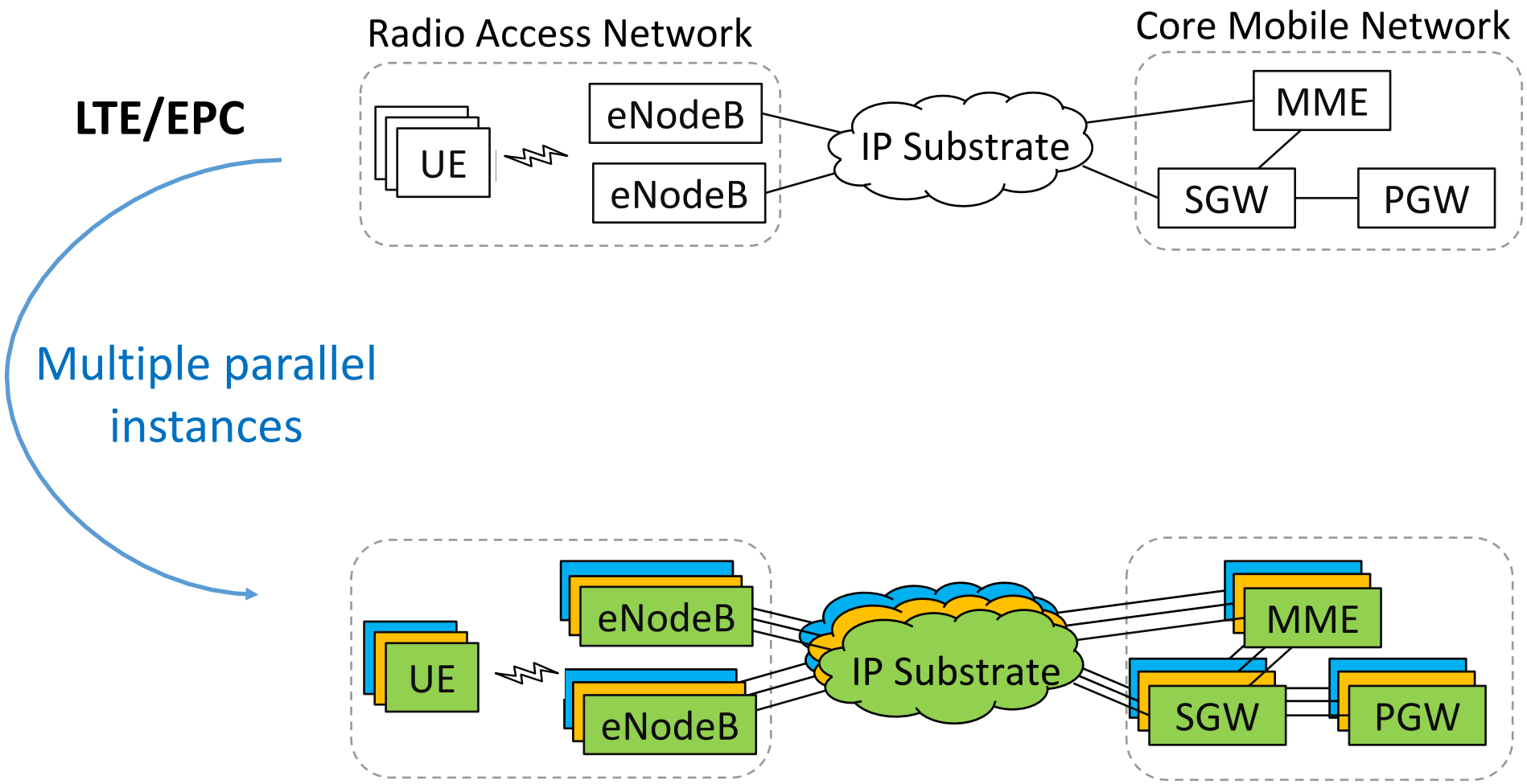
LTE/EPC: Standard mobile broadband service

Use Cases



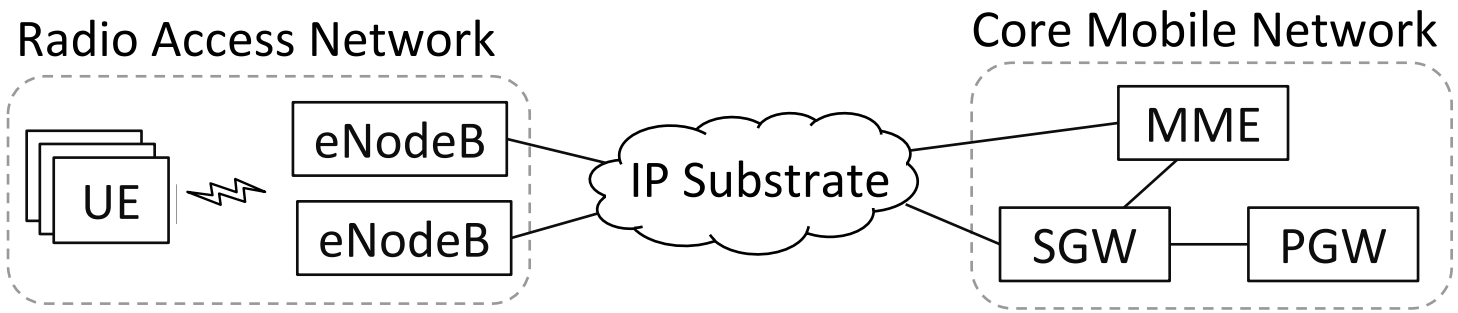
LTE/EPC: Standard mobile broadband service

Use Case: Hosted Multiplicity

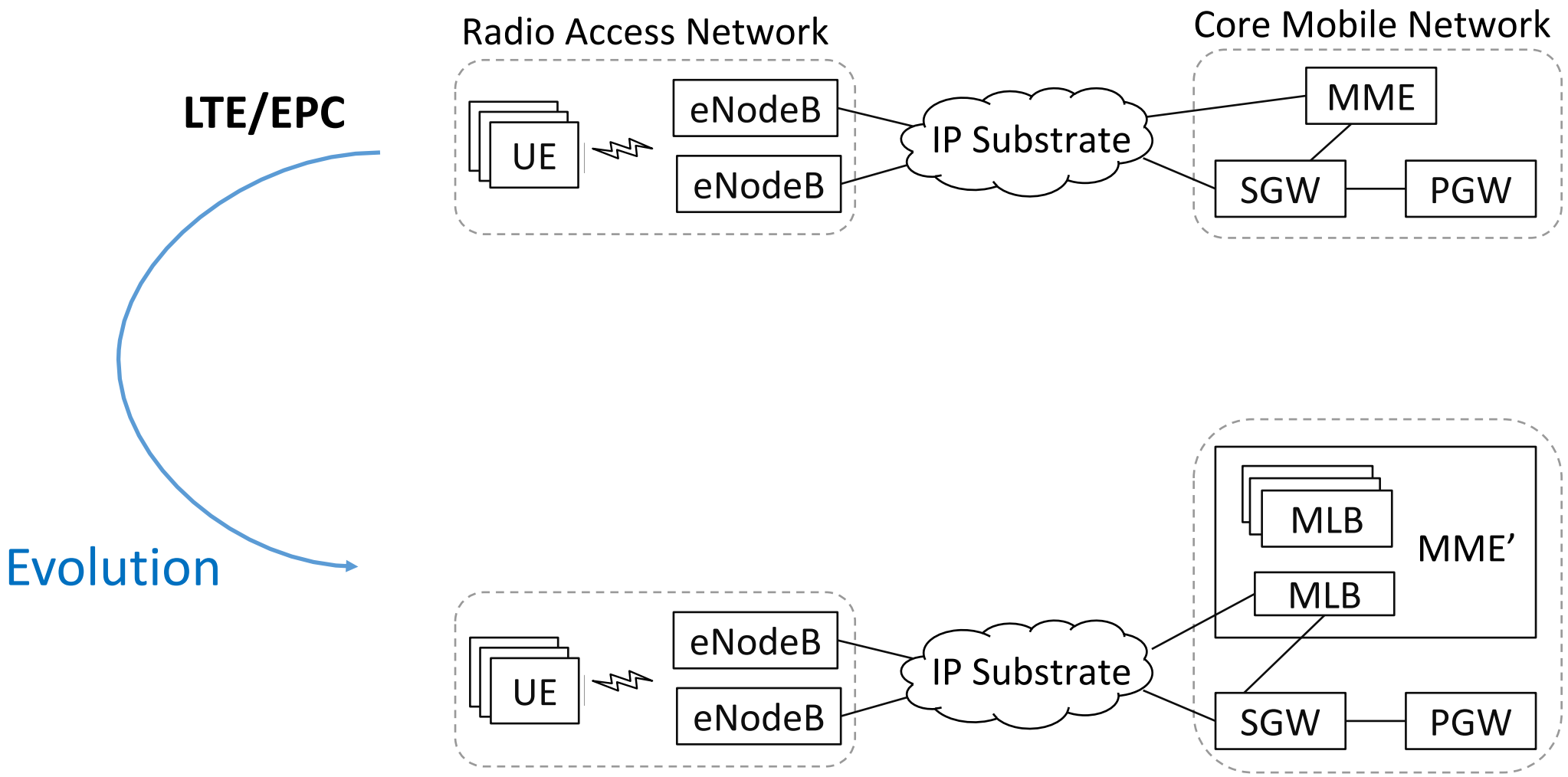


Use Case: SCALE

LTE/EPC

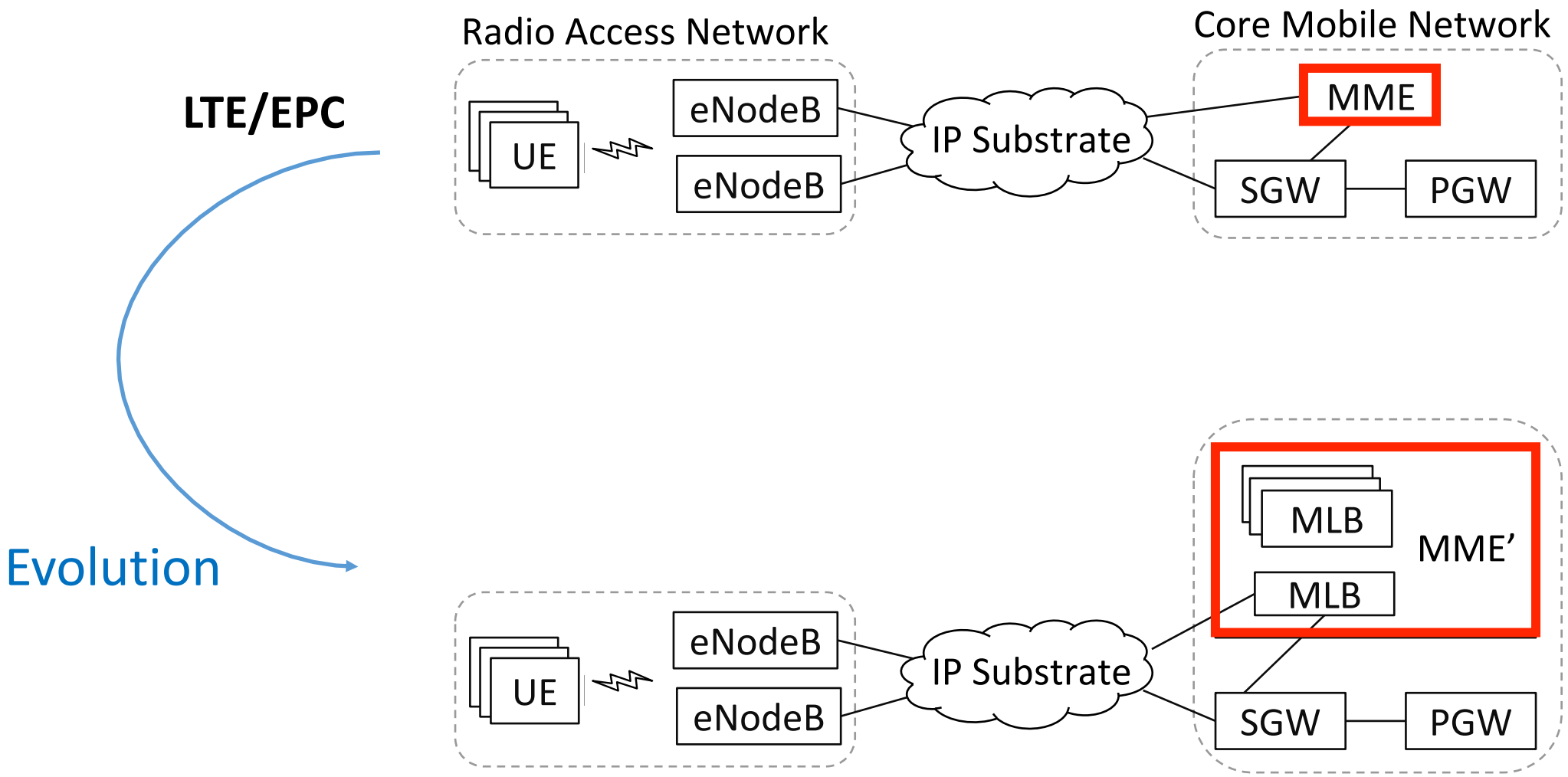


Use Case: SCALE



SCALE [Banerjee et al., CoNEXT 2015]

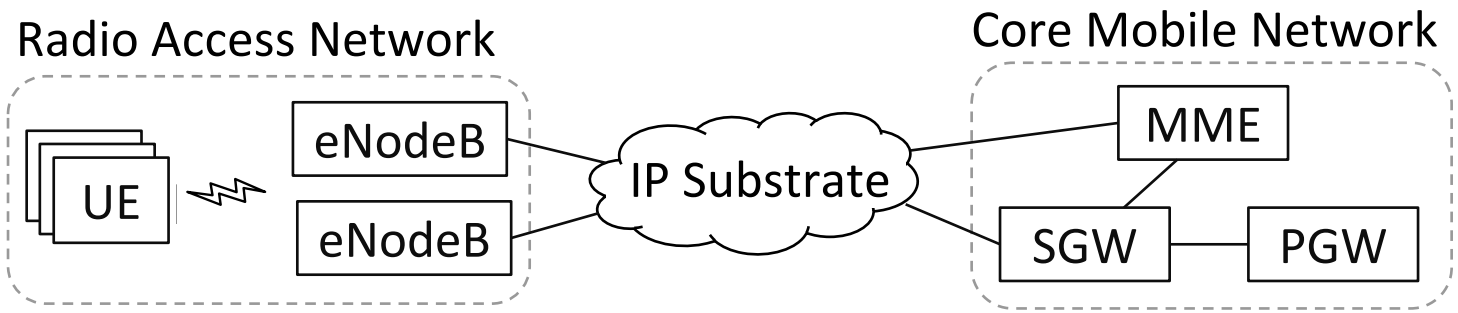
Use Case: SCALE



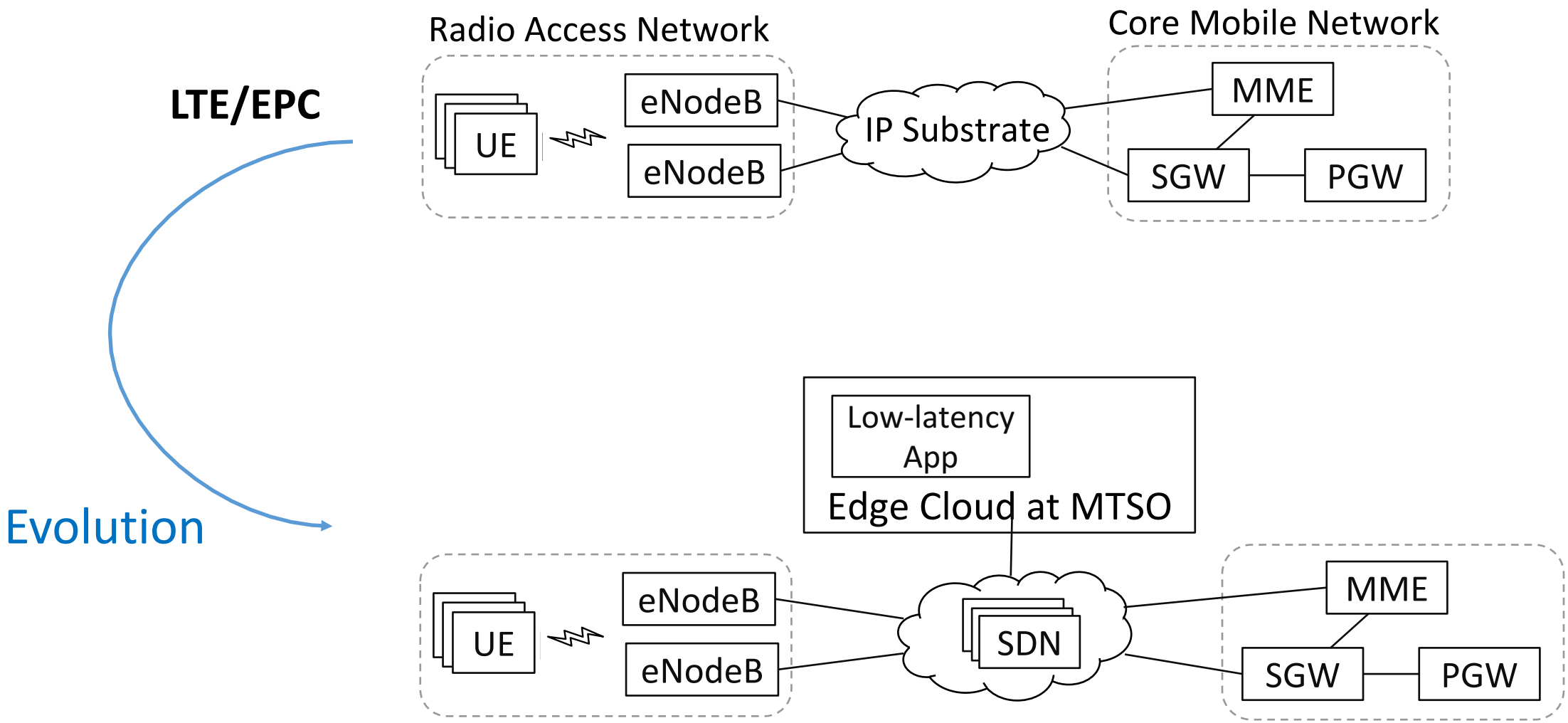
SCALE [Banerjee et al., CoNEXT 2015]

Use Case: SMORE

LTE/EPC

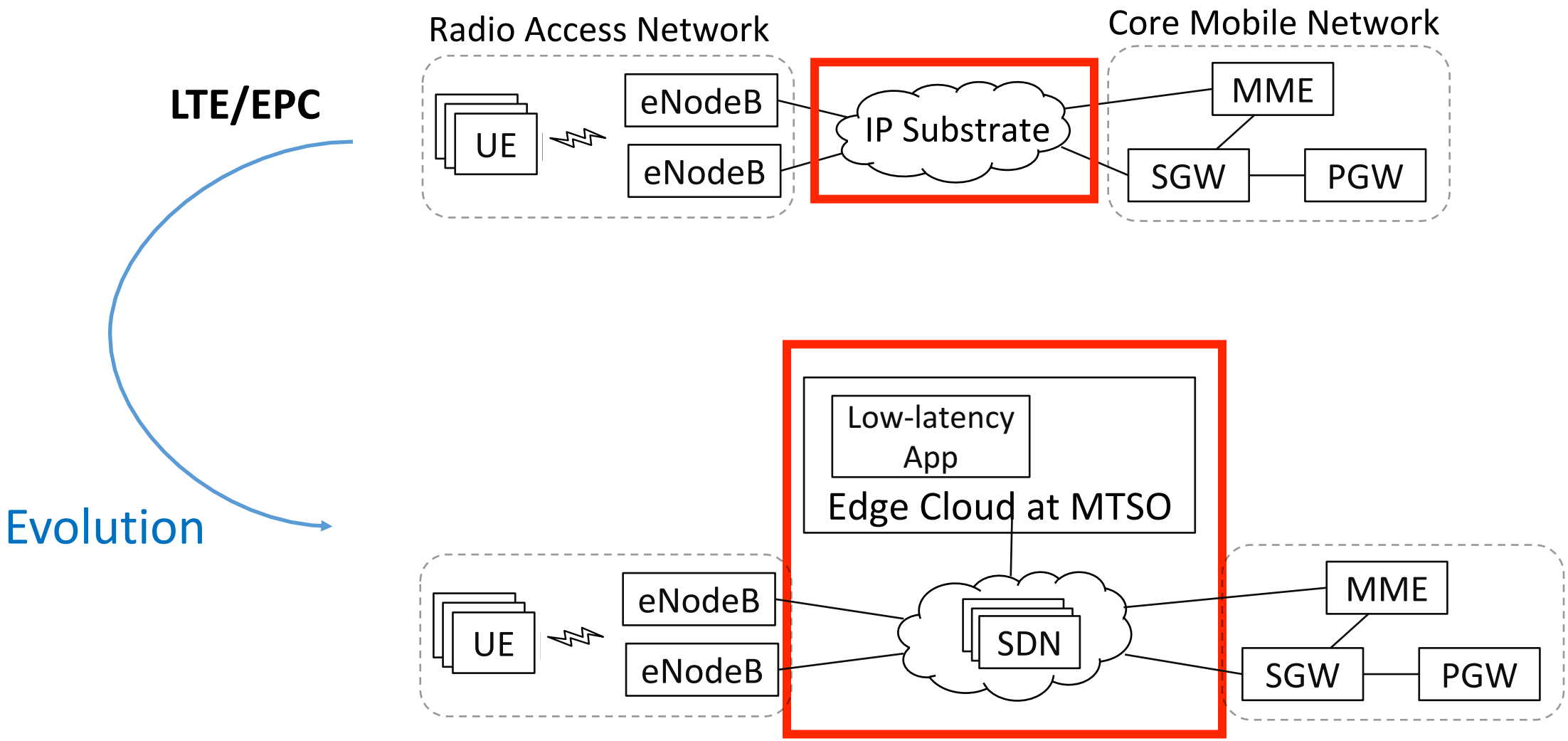


Use Case: SMORE



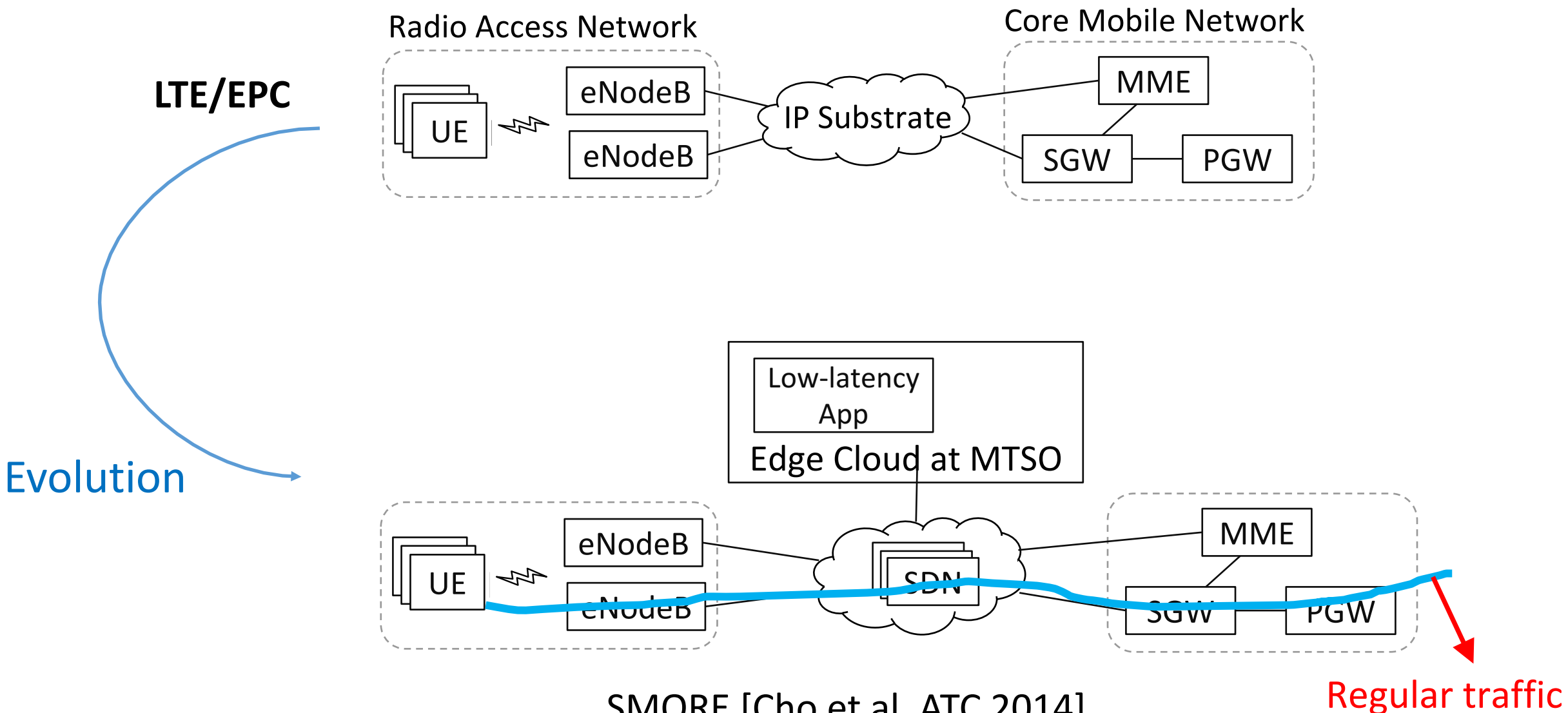
SMORE [Cho et al. ATC 2014]

Use Case: SMORE



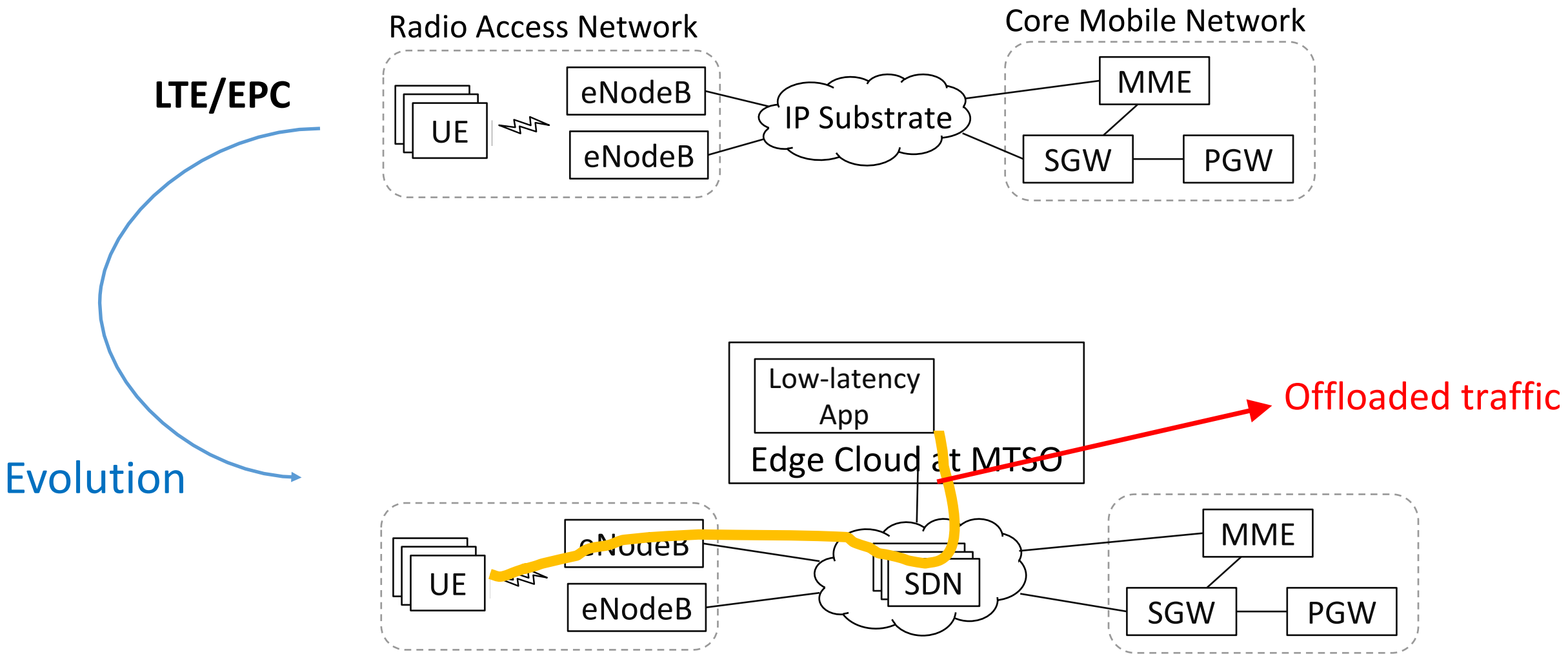
SMORE [Cho et al. ATC 2014]

Use Case: SMORE



SMORE [Cho et al. ATC 2014]

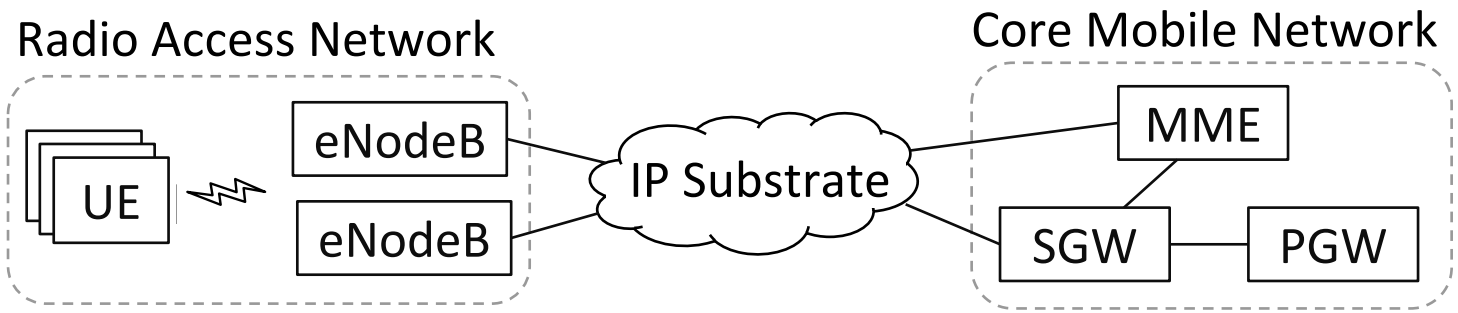
Use Case: SMORE



SMORE [Cho et al. ATC 2014]

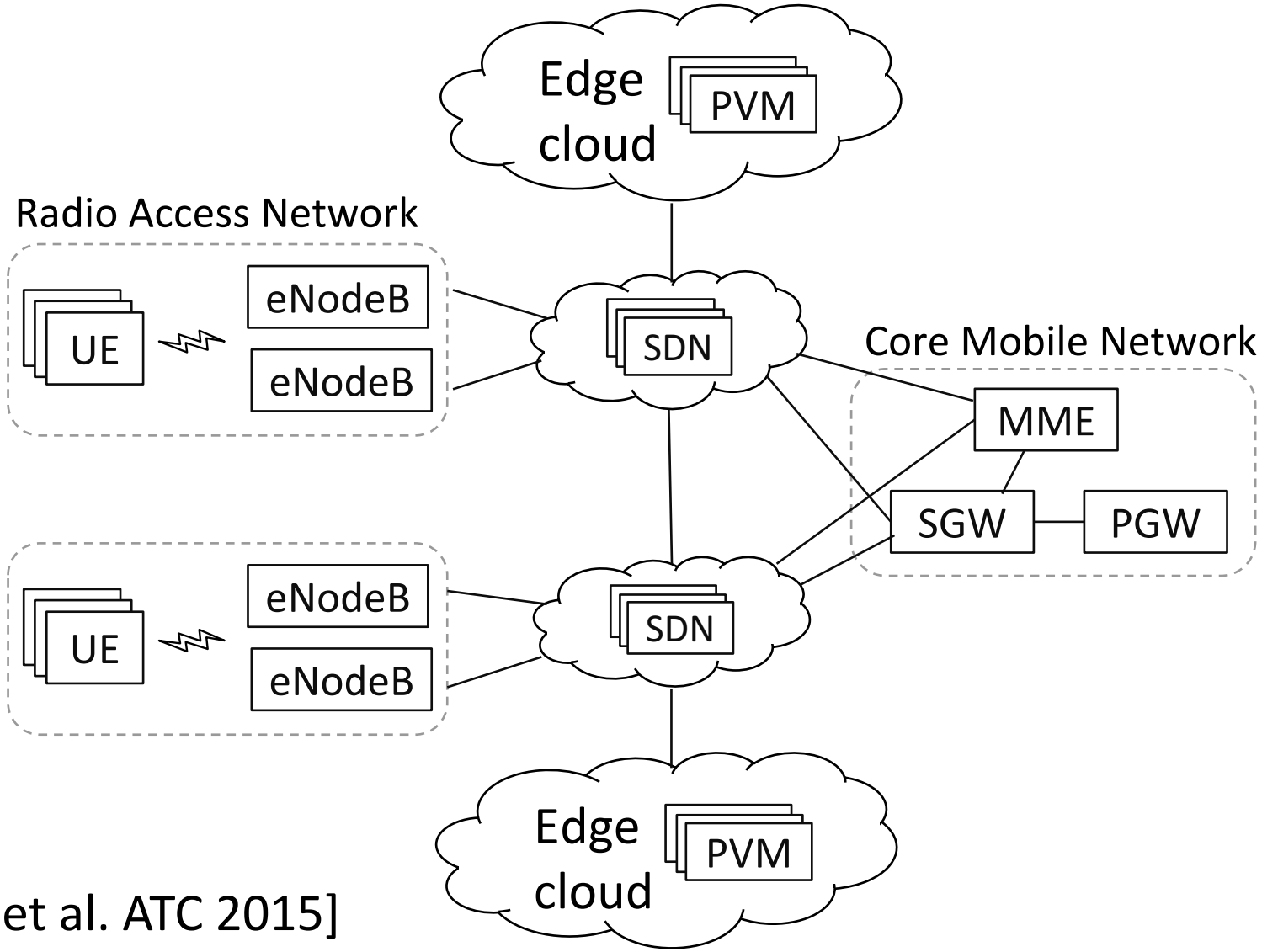
Use Case: MobiScud

LTE/EPC



Use Case: MobiScud

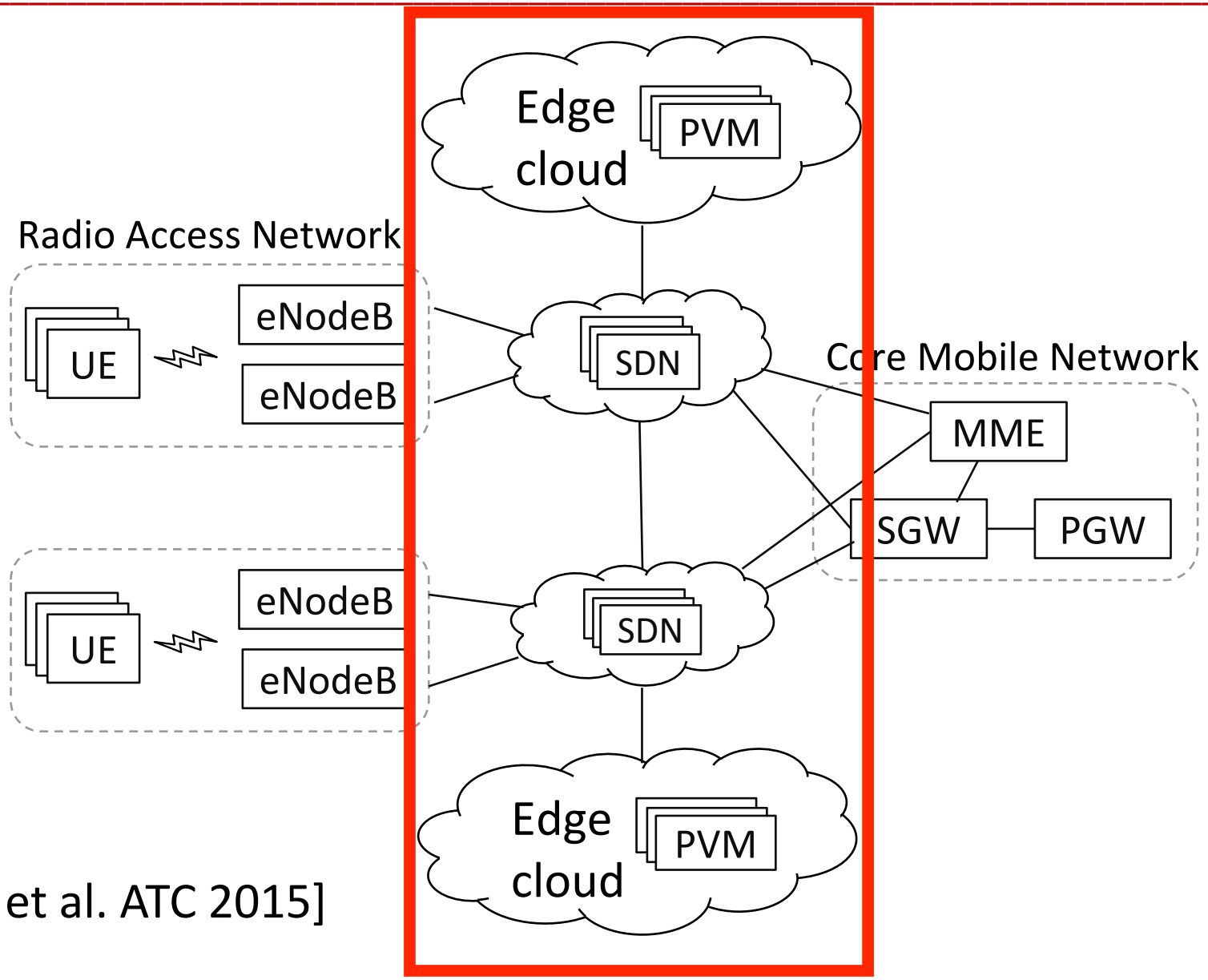
Evolution



MobiScud [Wang et al. ATC 2015]

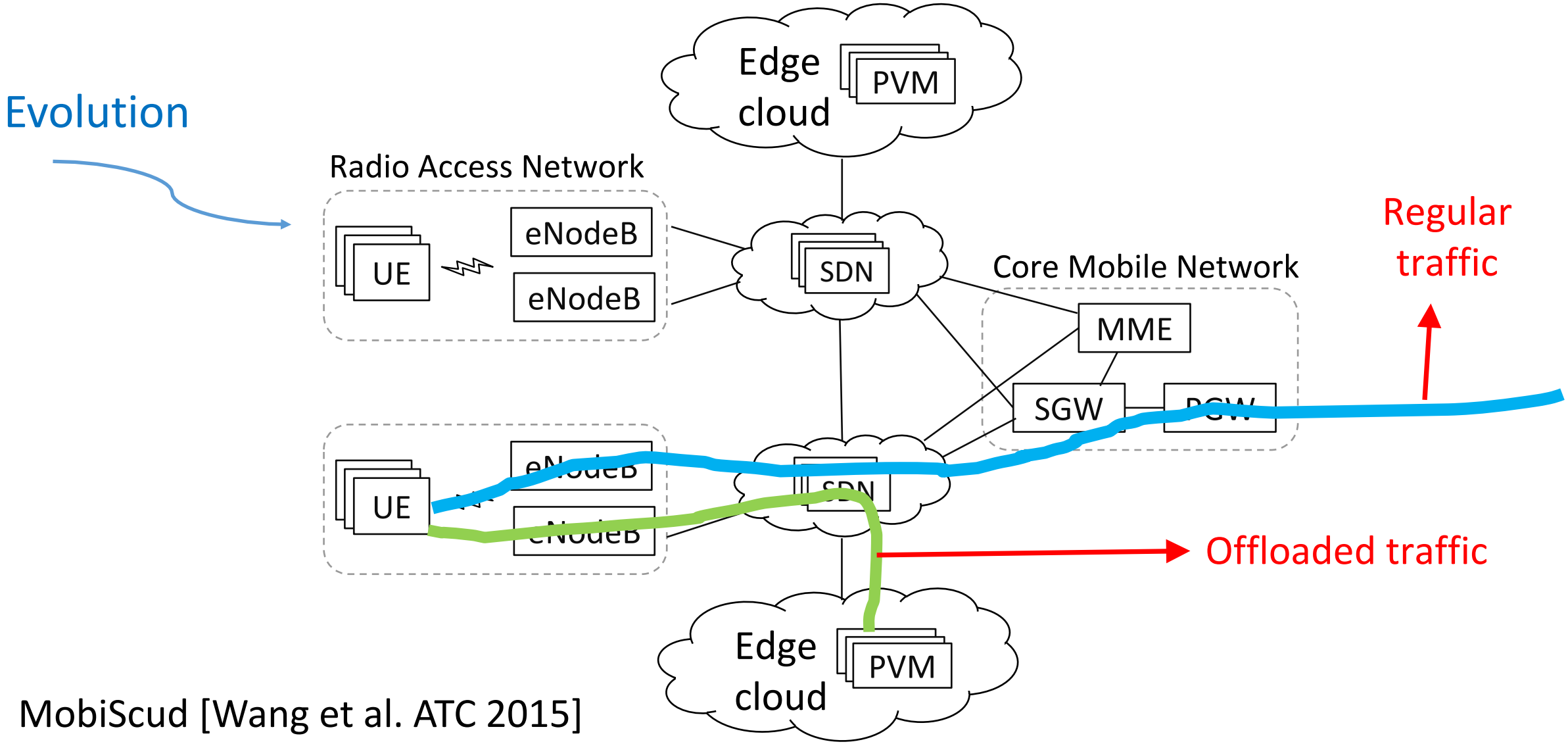
Use Case: MobiScud

Evolution



MobiScud [Wang et al. ATC 2015]

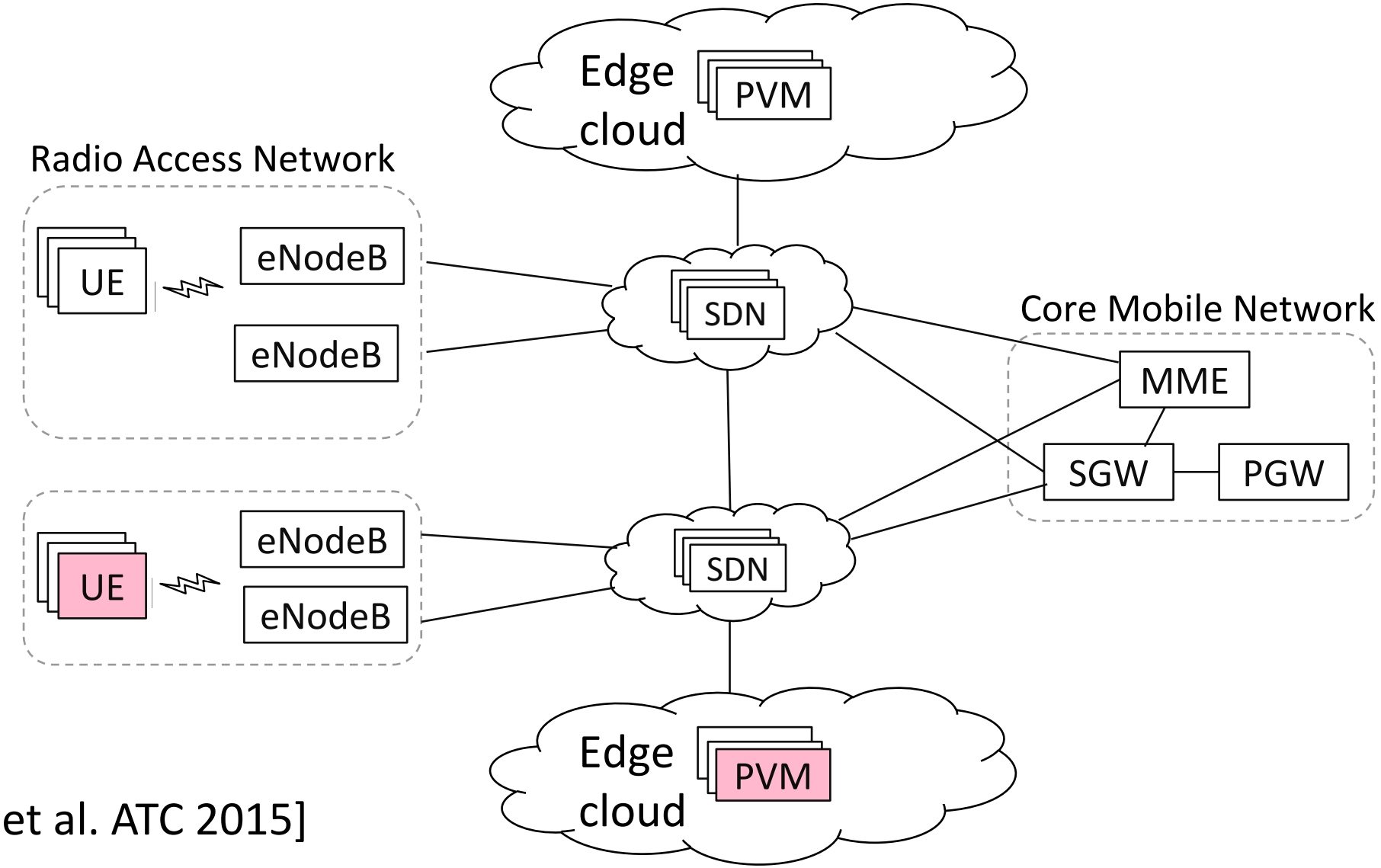
Use Case: MobiScud



MobiScud [Wang et al. ATC 2015]

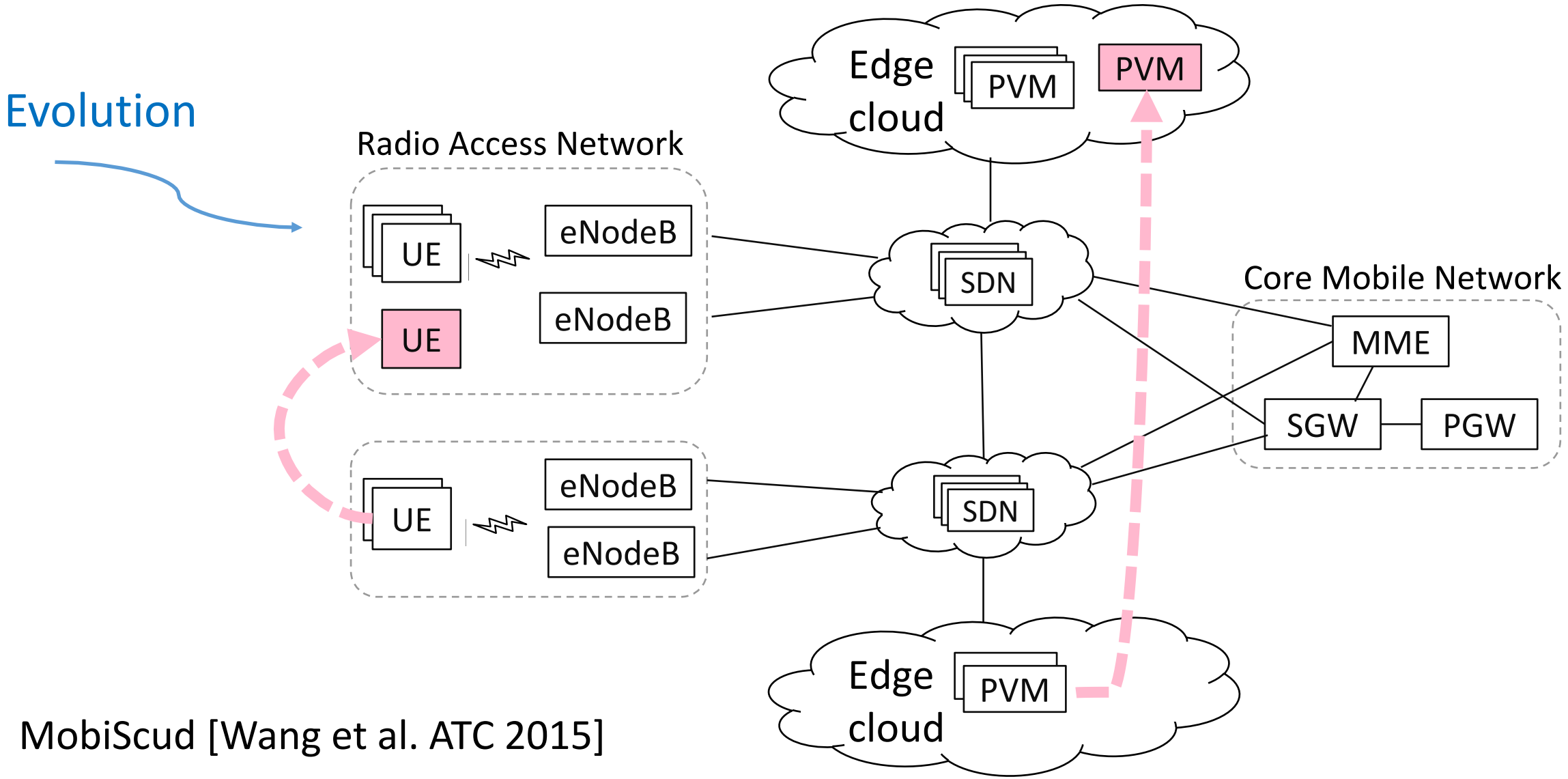
Use Case: MobiScud

Evolution



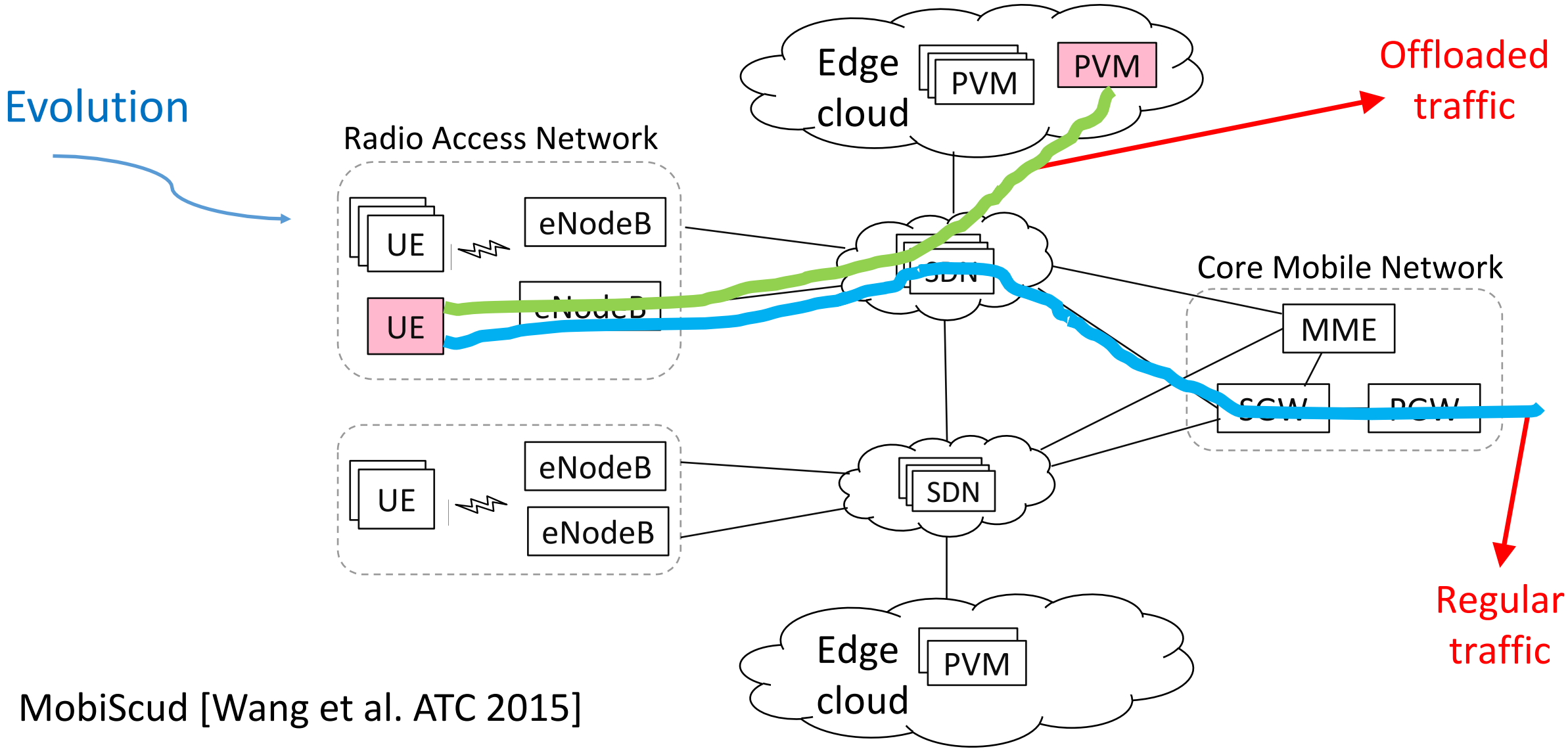
MobiScud [Wang et al. ATC 2015]

Use Case: MobiScud



MobiScud [Wang et al. ATC 2015]

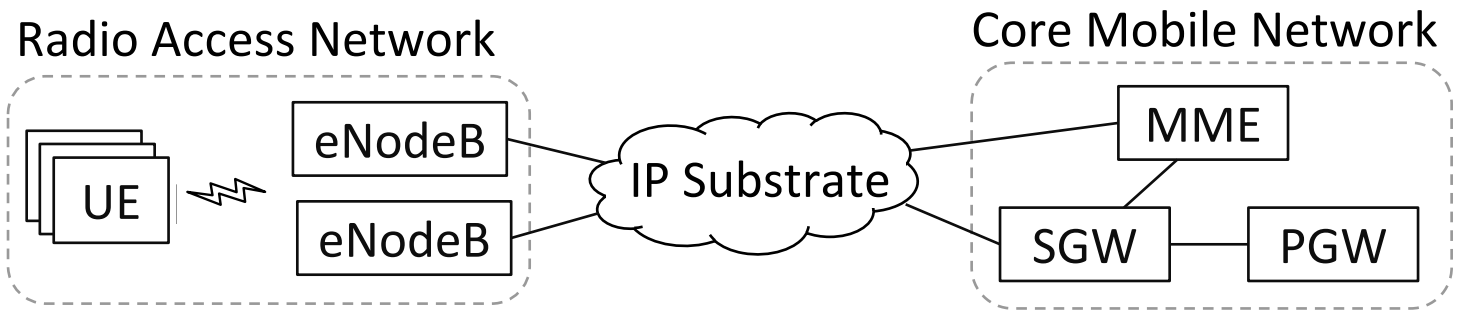
Use Case: MobiScud



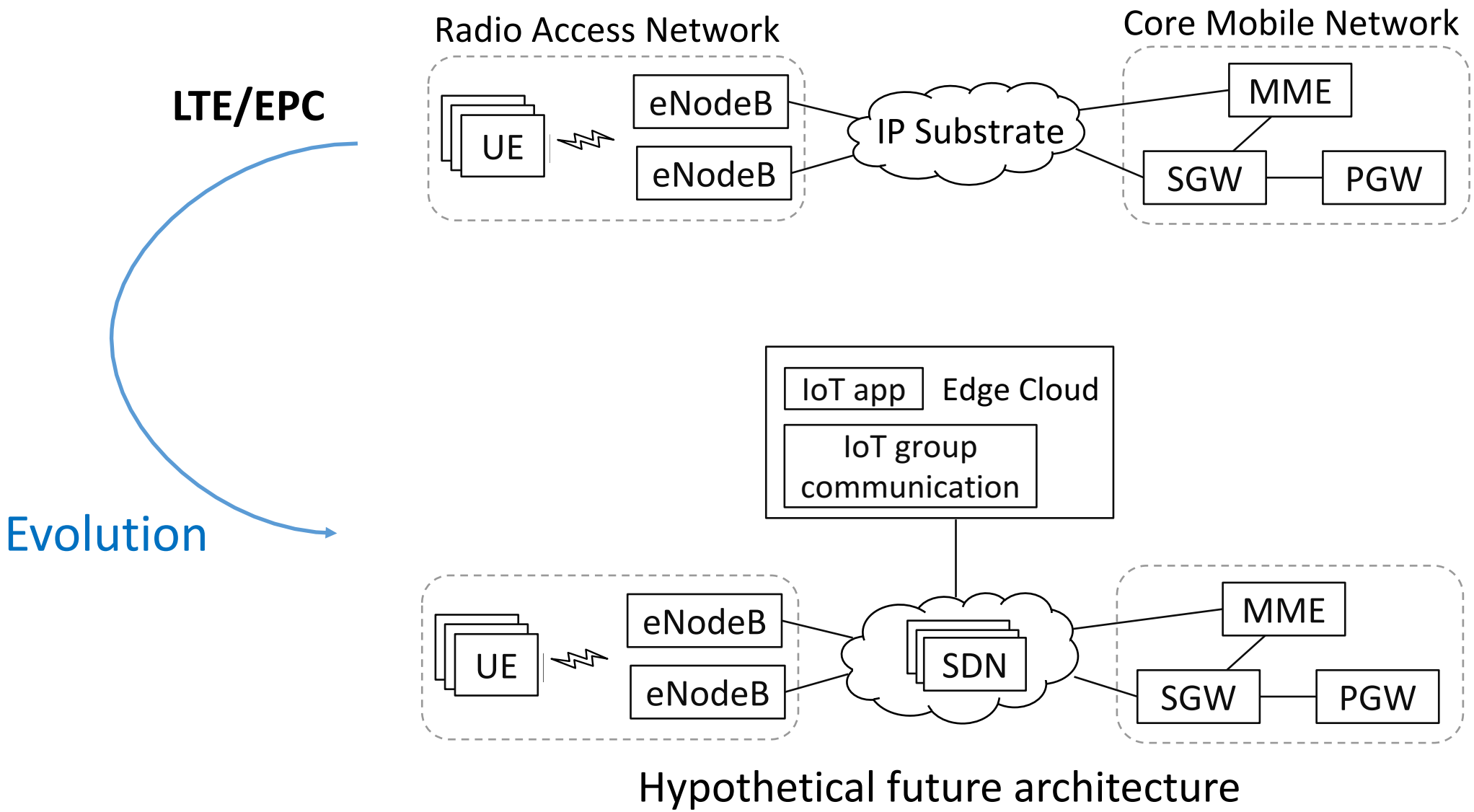
MobiScud [Wang et al. ATC 2015]

Use Case: Specialized EPC for IoT

LTE/EPC

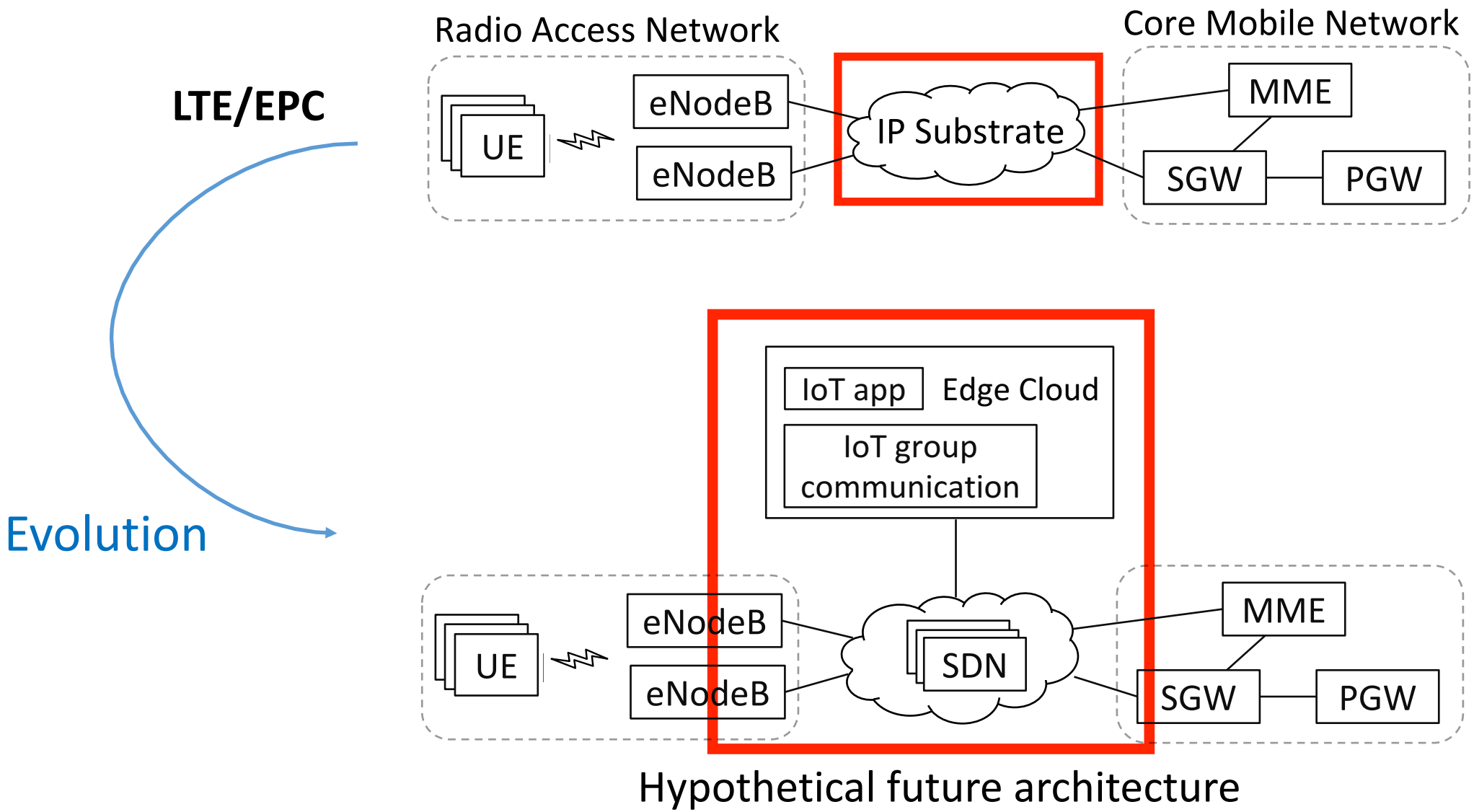


Use Case: Specialized EPC for IoT



Hypothetical future architecture

Use Case: Specialized EPC for IoT



Hypothetical future architecture

Use Cases

- 4G LTE/EPC
- Multiple parallel instances
- SCALE (MME replacement with a more elastic version)
- SMORE (selective low-latency offloading)
- MobiScud (selective offloading with PVMs)
- IoT-specific mobile architecture

Use Cases

- 4G LTE/EPC
- Multiple parallel instances
- SCALE (MME replacement with a more elastic version)
- SMORE (selective low-latency offloading)
- MobiScud (selective offloading with PVMs)
- IoT-specific mobile architecture

Proteus

- Proteus enables

Proteus

- Proteus enables
  - safe, rapid, dynamic service evolution

Proteus

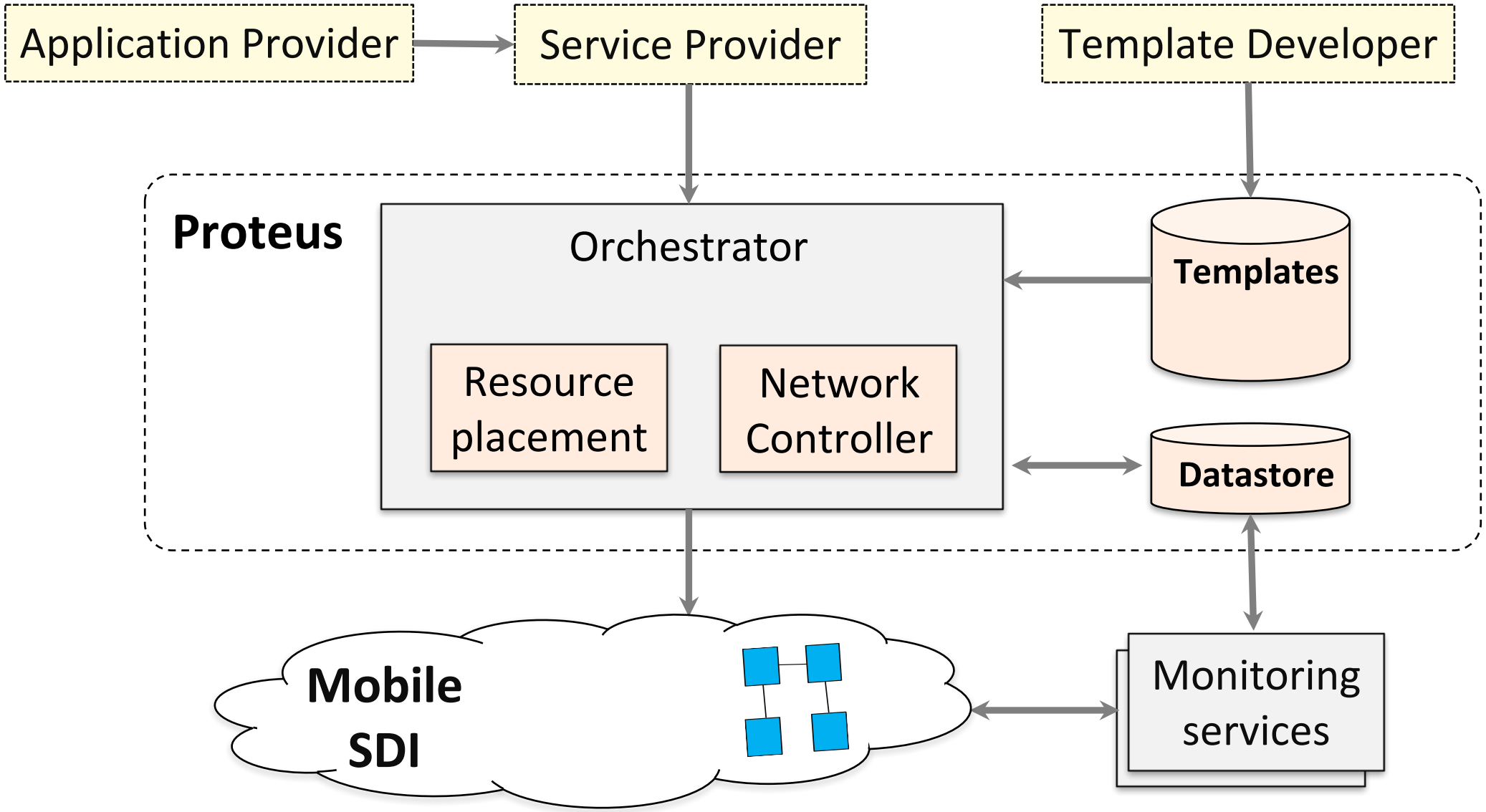
- Proteus enables
  - safe, rapid, dynamic service evolution
  - multiple mobile service instances in parallel

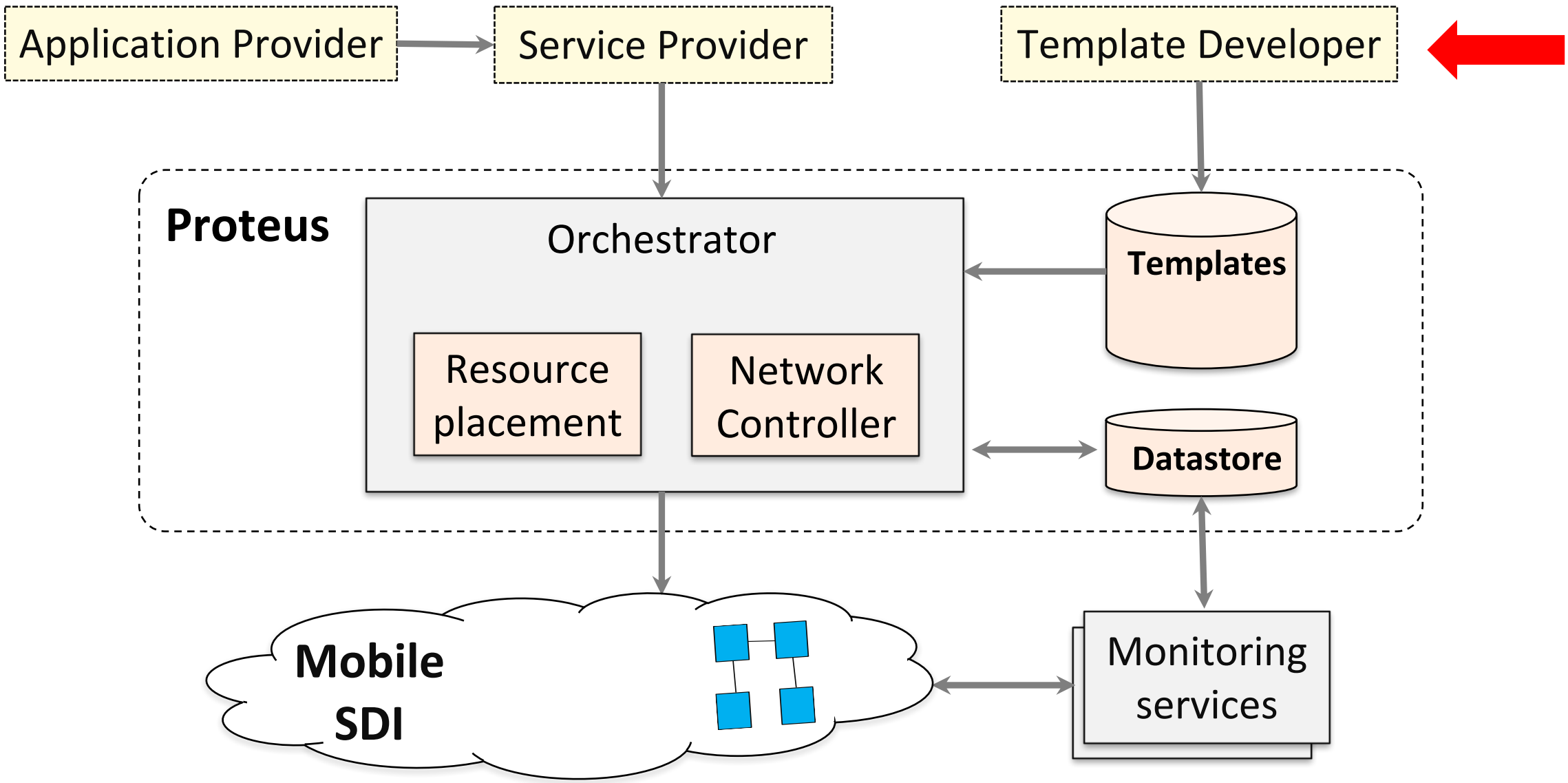
Proteus

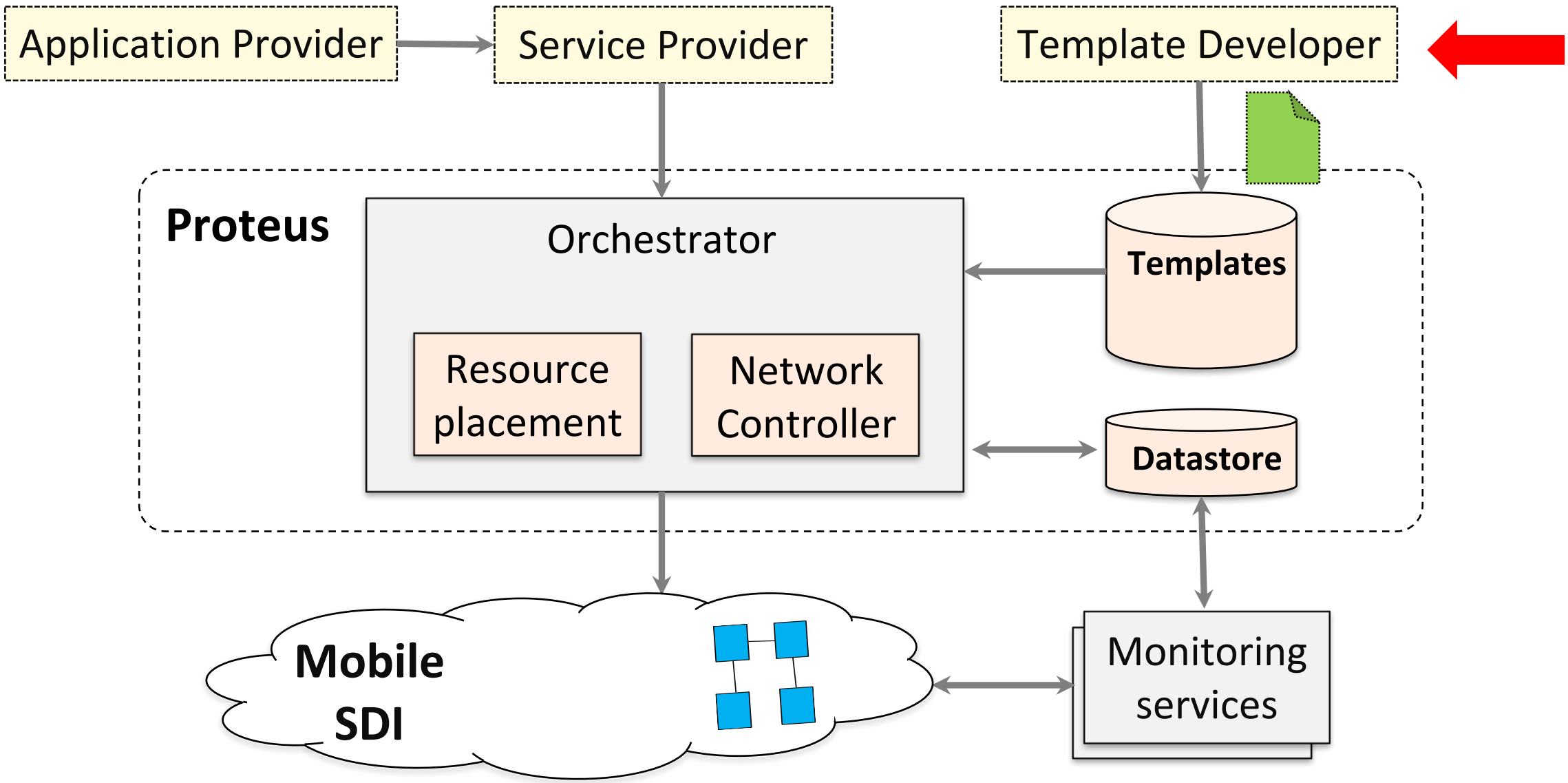
- Proteus enables
  - safe, rapid, dynamic service evolution
  - multiple mobile service instances in parallel
  - cloud-like mobile network abstractions

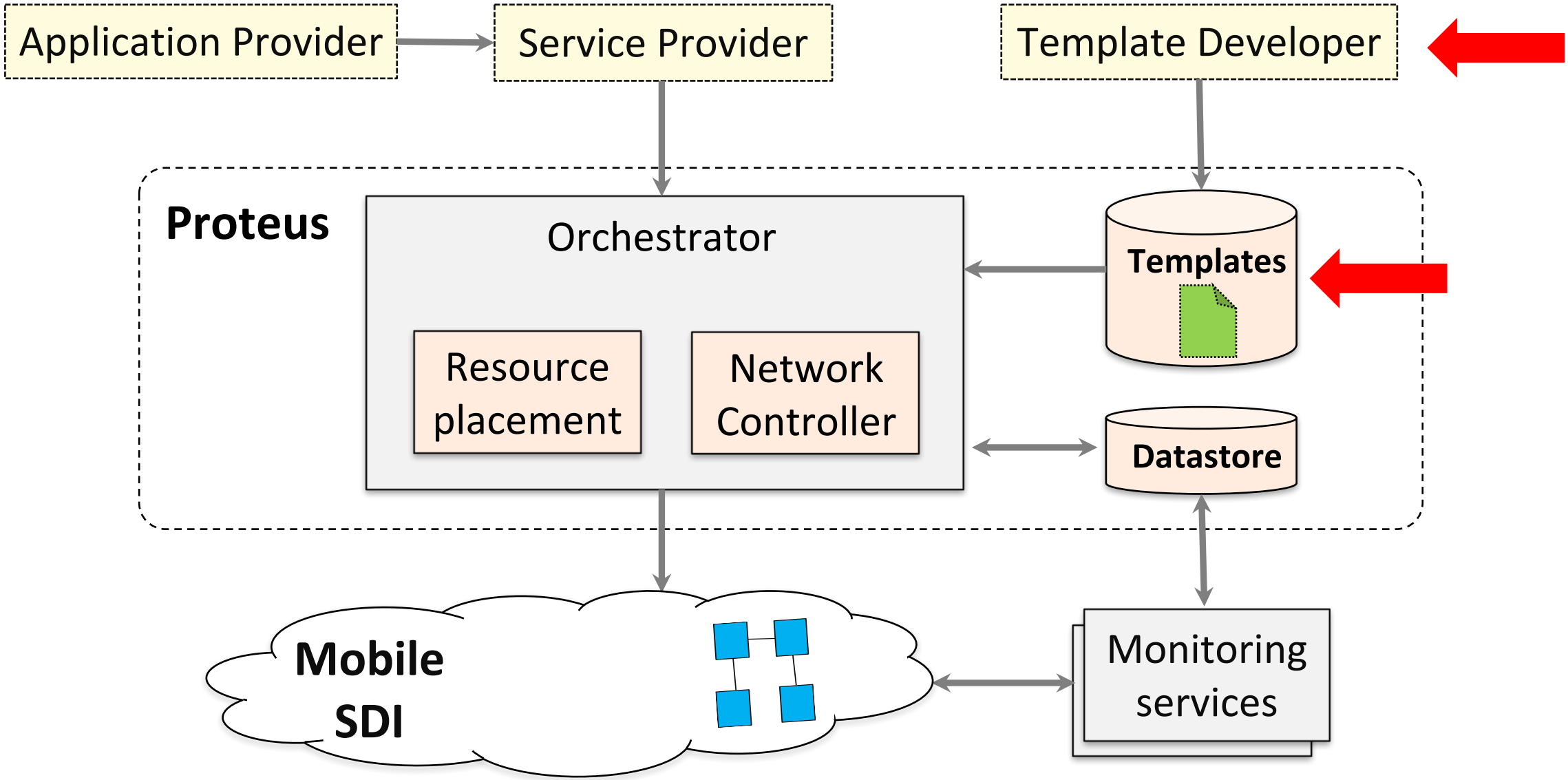
Proteus

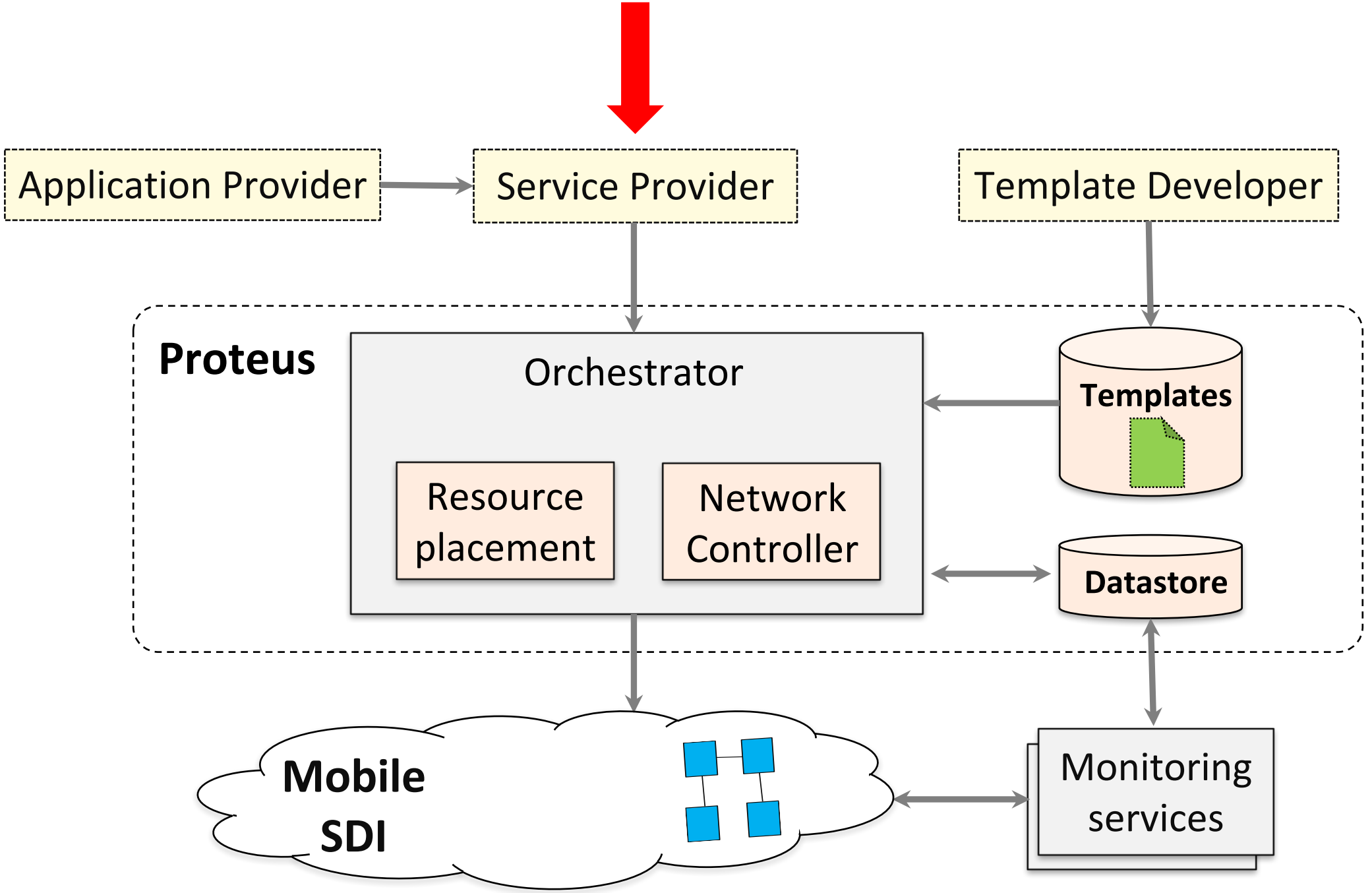
- Proteus enables
  - safe, rapid, dynamic service evolution
  - multiple mobile service instances in parallel
  - cloud-like mobile network abstractions
  - single mobile platform to be opened up to third-party providers

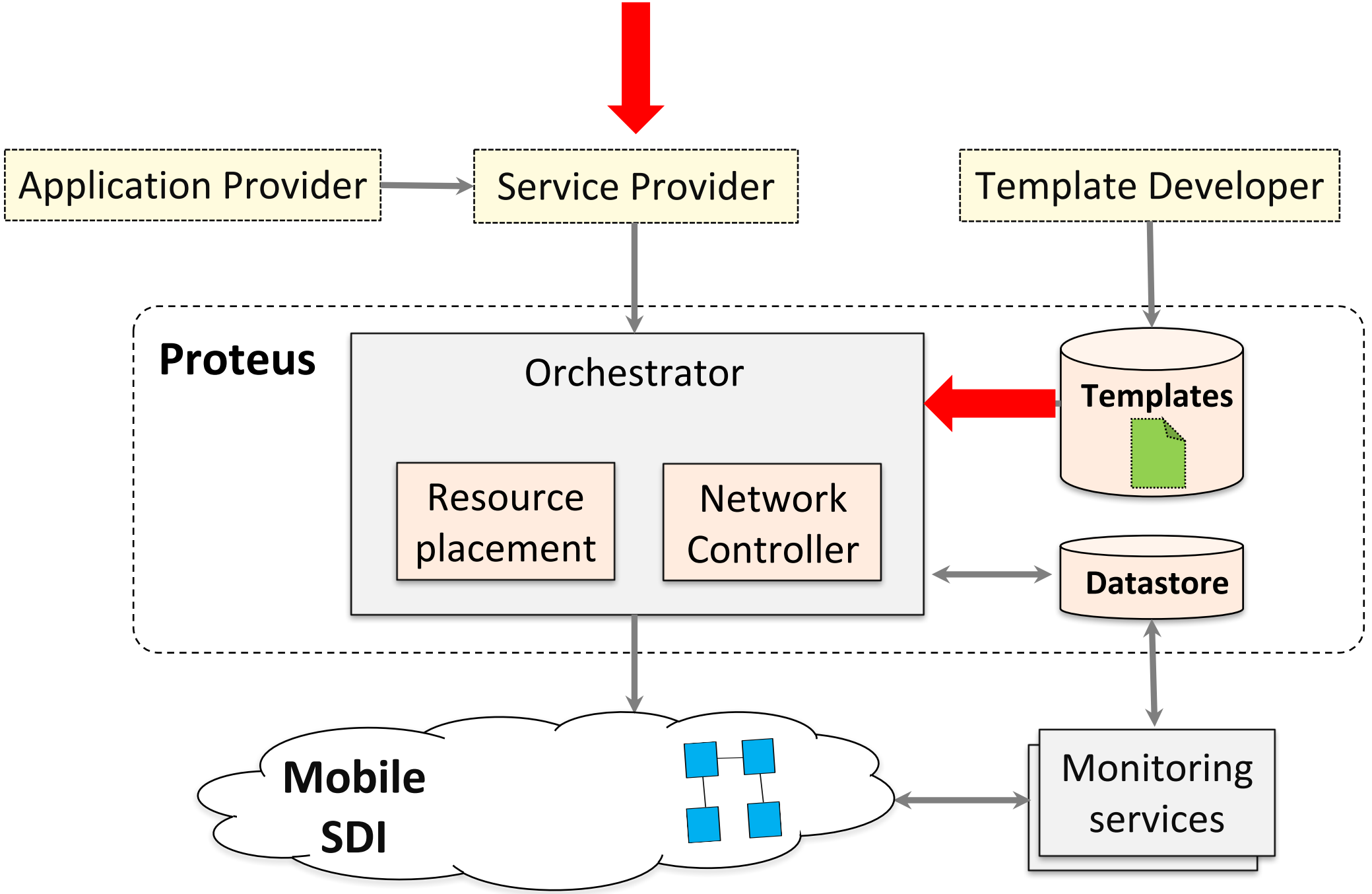

Application Provider
Service Provider
Template Developer
Proteus
Orchestrator
Resource placement
Network Controller
Templates
Datastore
Mobile SDI
Monitoring services

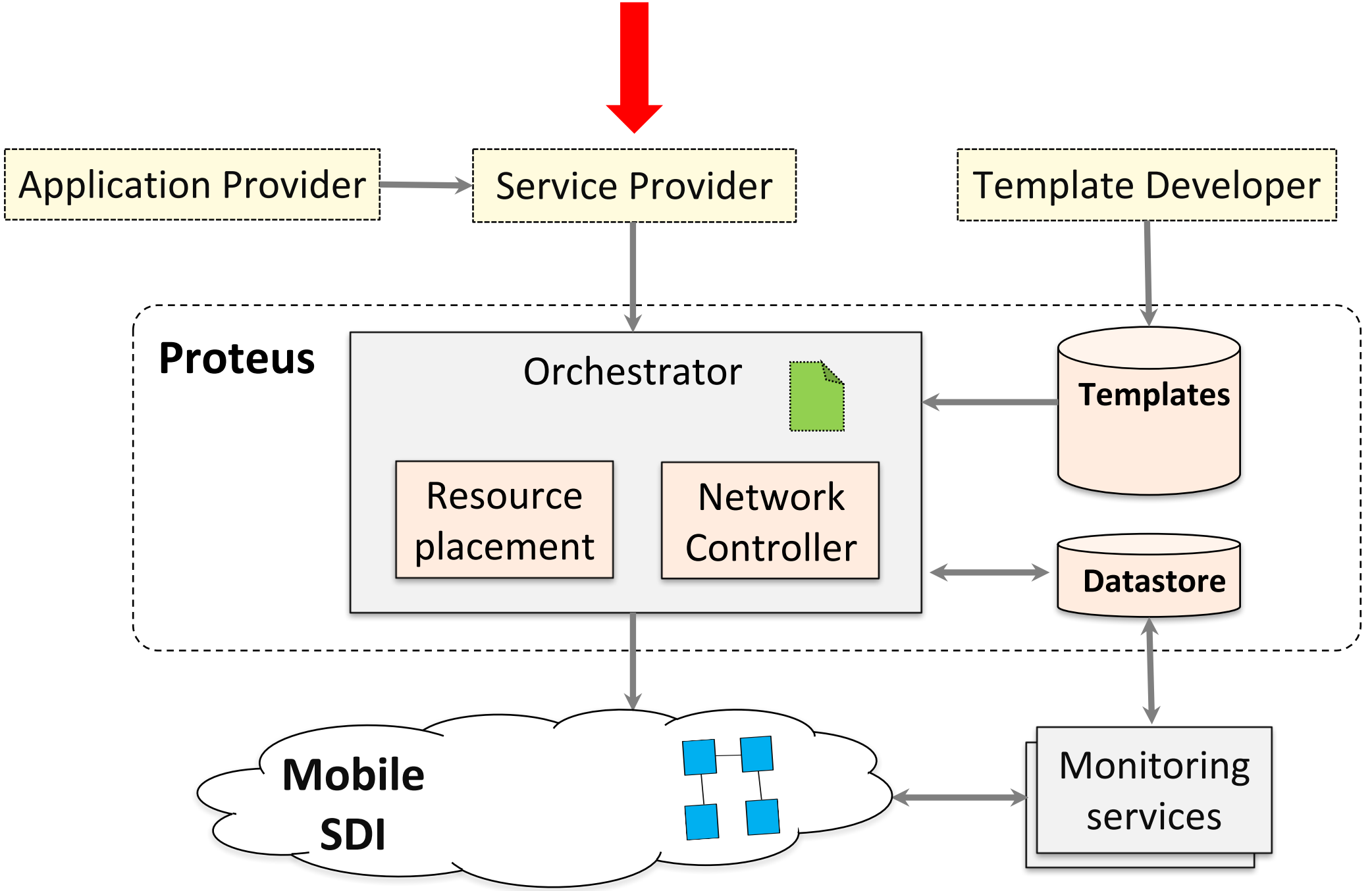

Application Provider
Service Provider
Template Developer
Proteus
Orchestrator
Resource placement
Network Controller
Templates
Datastore
Mobile SDI
Monitoring services

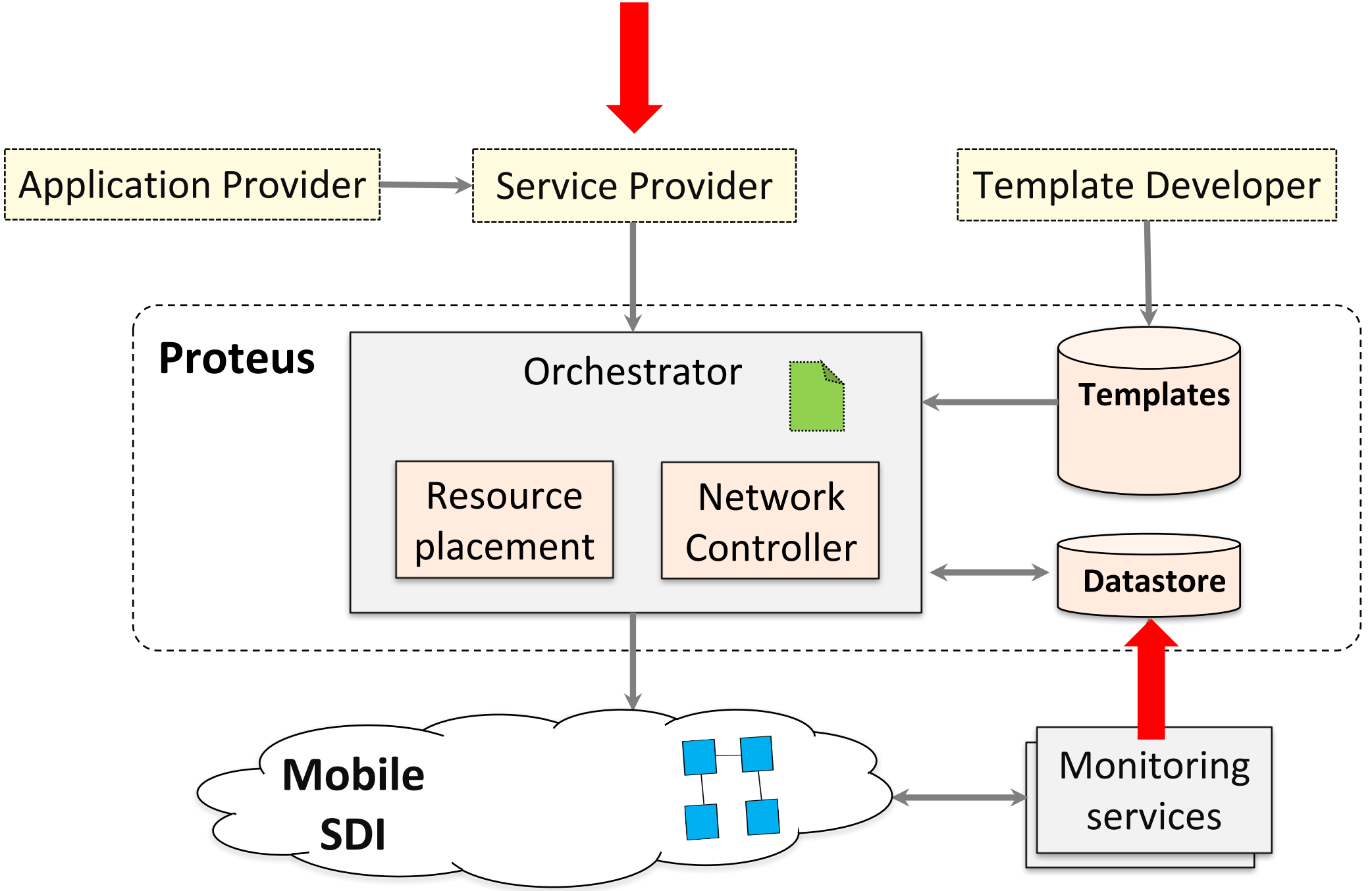

Application Provider
Service Provider
Template Developer
Proteus
Orchestrator
Resource placement
Network Controller
Templates
Datastore
Mobile SDI
Monitoring services

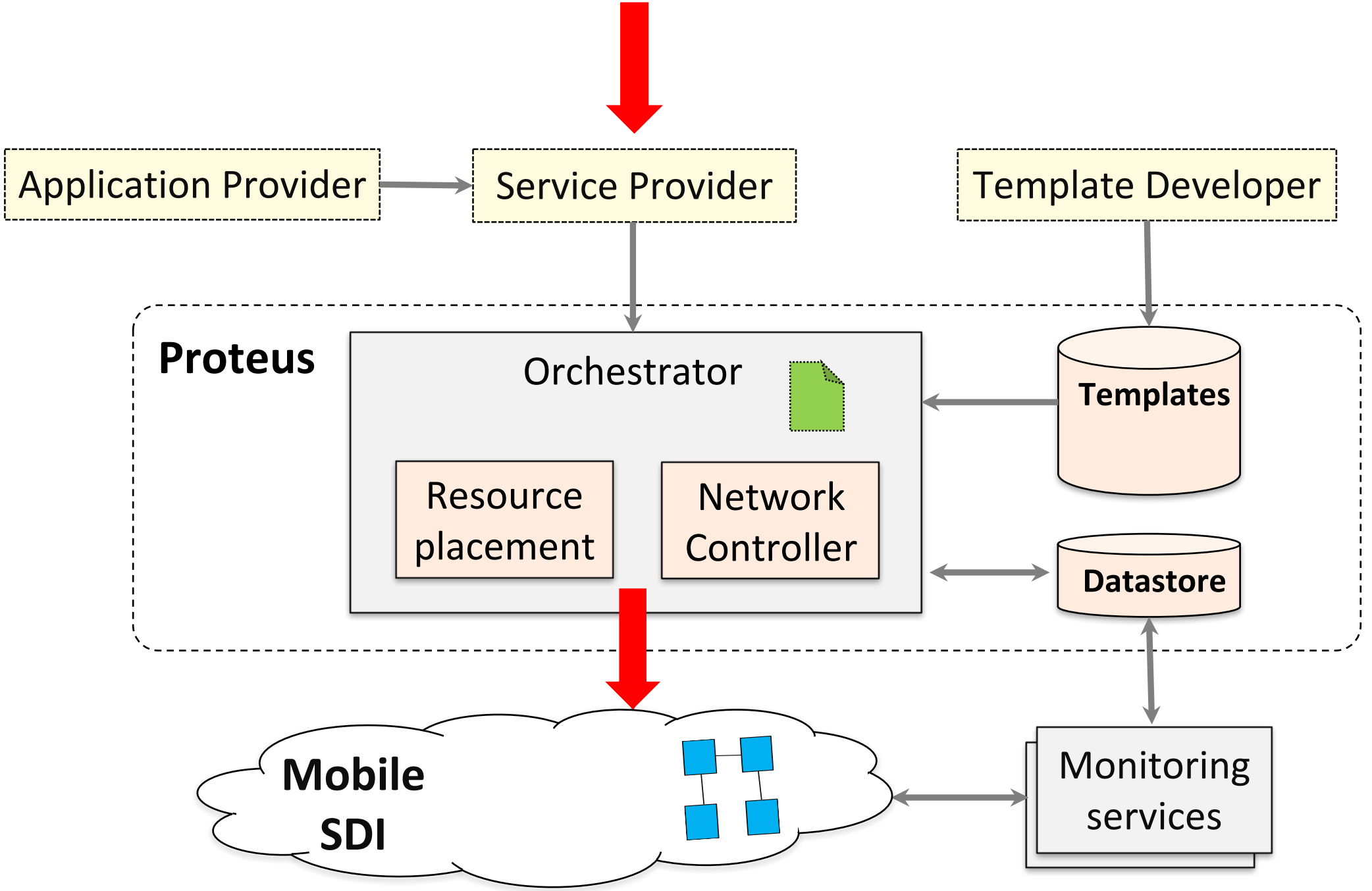

Application Provider
Service Provider
Template Developer
Proteus
Orchestrator
Resource placement
Network Controller
Templates
Datastore
Mobile SDI
Monitoring services

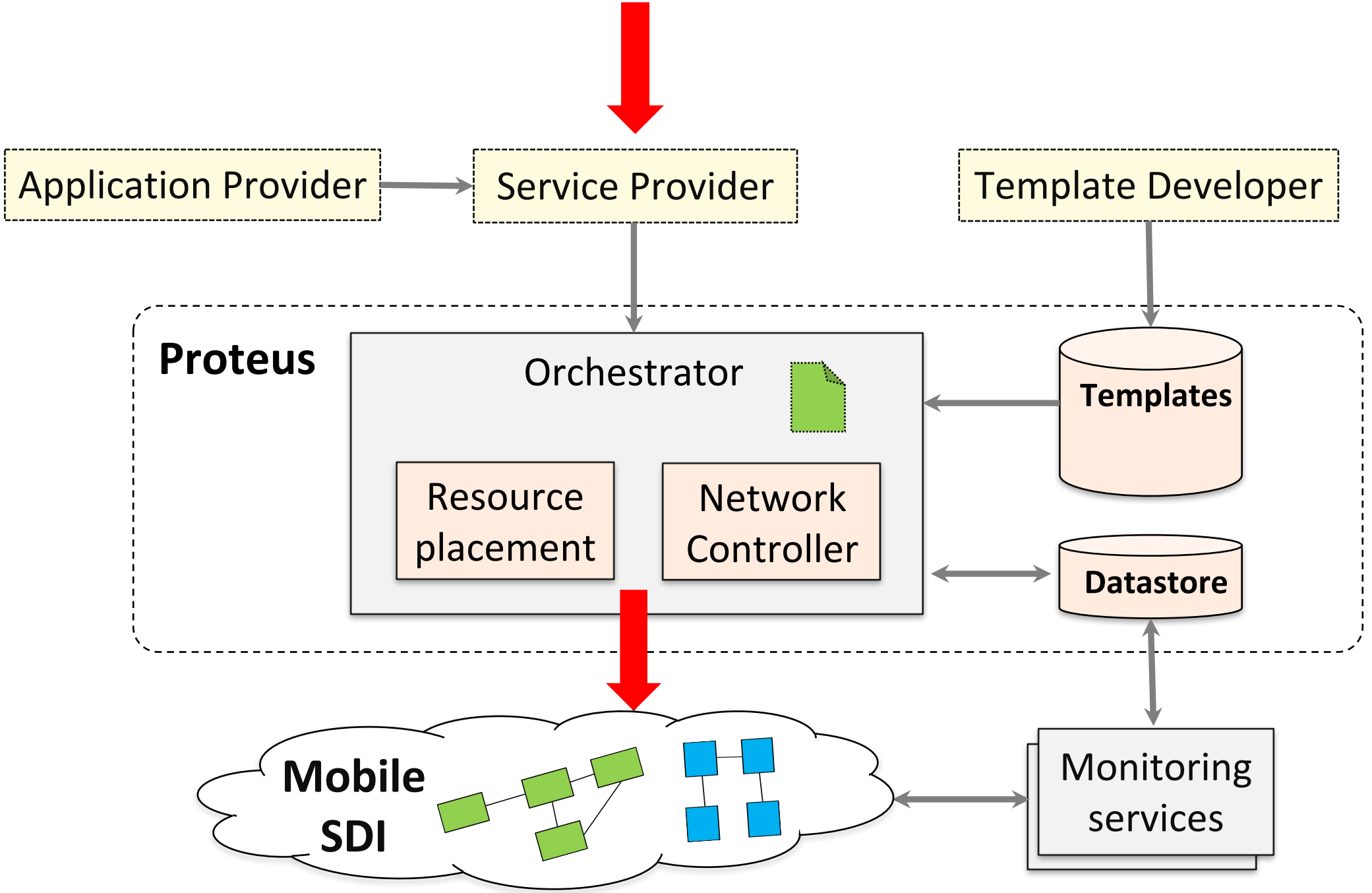

Application Provider
Service Provider
Template Developer
Proteus
Orchestrator
Resource placement
Network Controller
Templates
Datastore
Mobile SDI
Monitoring services

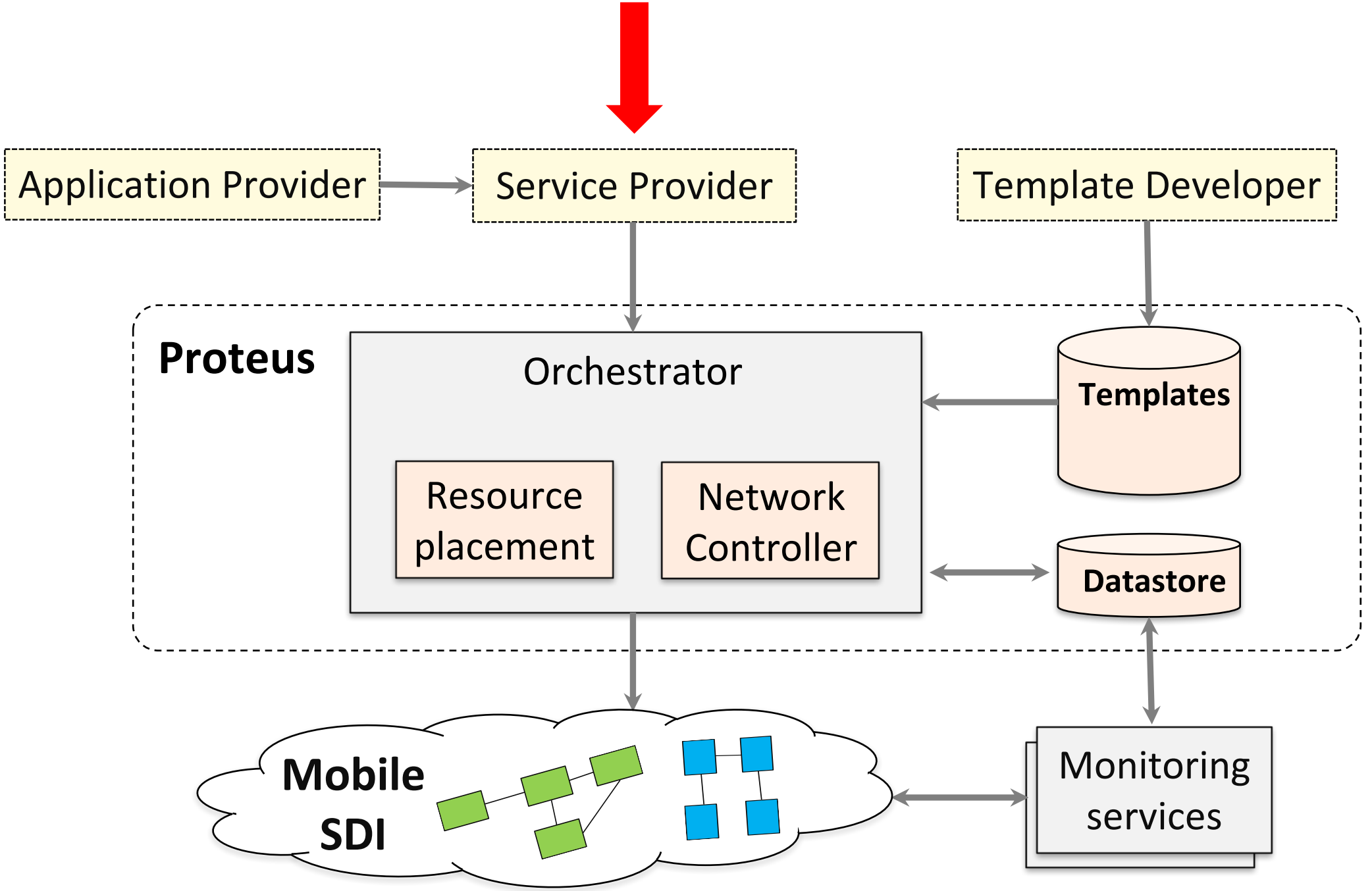

Application Provider
Service Provider
Template Developer
Proteus
Orchestrator
Resource placement
Network Controller
Templates
Datastore
Mobile SDI
Monitoring services


Application Provider
Service Provider
Template Developer
Proteus
Orchestrator
Resource placement
Network Controller
Templates
Datastore
Mobile SDI
Monitoring services

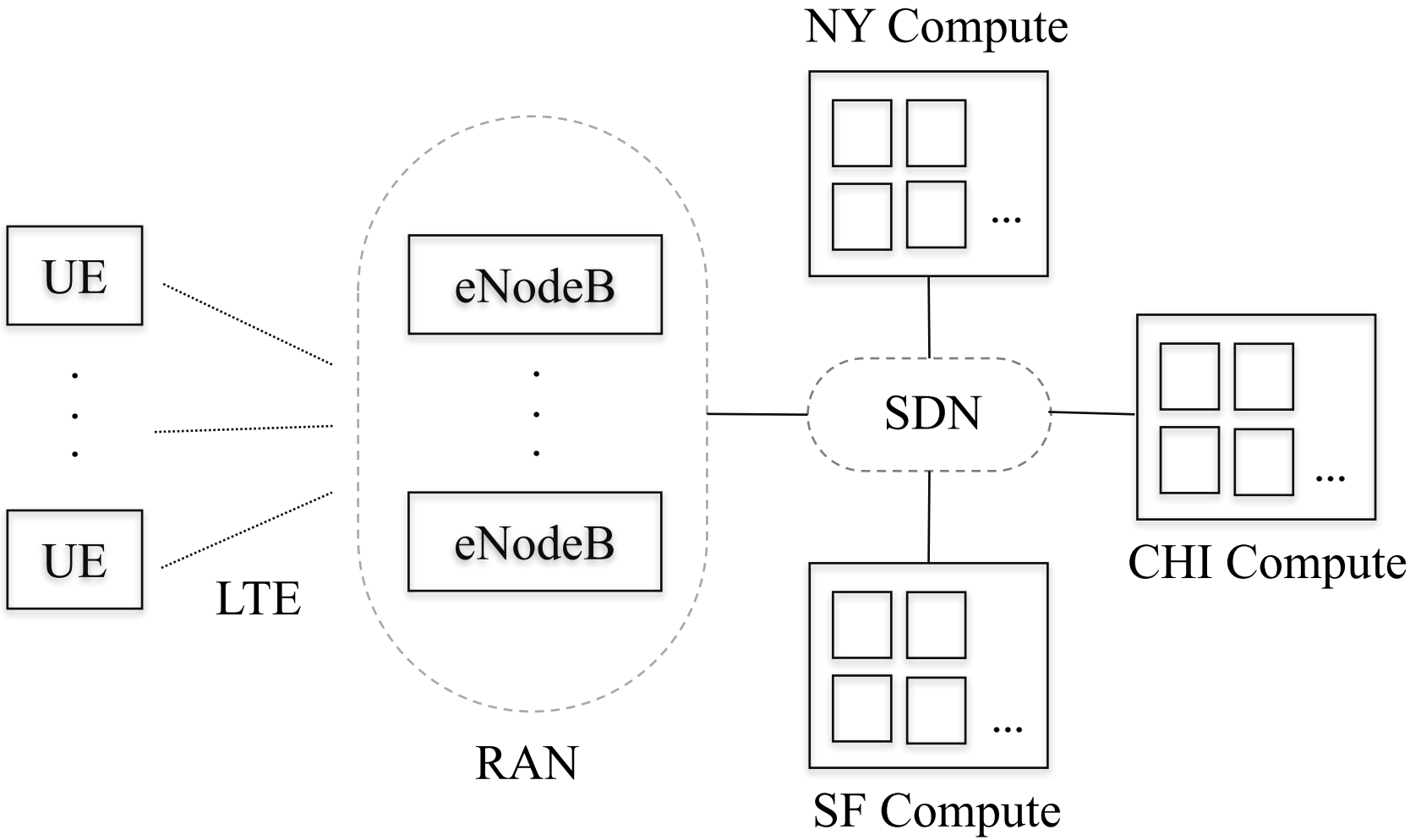

Application Provider
Service Provider
Template Developer
Proteus
Orchestrator
Resource placement
Network Controller
Templates
Datastore
Mobile SDI
Monitoring services


Application Provider
Service Provider
Template Developer
Proteus
Orchestrator
Resource placement
Network Controller
Templates
Datastore
Mobile SDI
Monitoring services


Application Provider
Service Provider
Template Developer
Proteus
Orchestrator
Resource placement
Network Controller
Templates
Datastore
Mobile SDI
Monitoring services


Application Provider
Service Provider
Template Developer
Proteus
Orchestrator
Resource placement
Network Controller
Templates
Datastore
Mobile SDI
Monitoring services

Evaluations



Topology in PhantomNet mobility testbed

Evaluations

Evaluations

- Service evolution and parallel instances
  - LTE/EPC
  - EPC evolved with SMORE
  - EPC evolved with MobiScud
- Data-centric service management
  - Handling bursts of UE attachments
  - Migrating to a new location
- Scalability
  - End-to-end orchestration request completion time
  - Orchestration time for individual components
  - End-to-end parallel orchestration request completion time

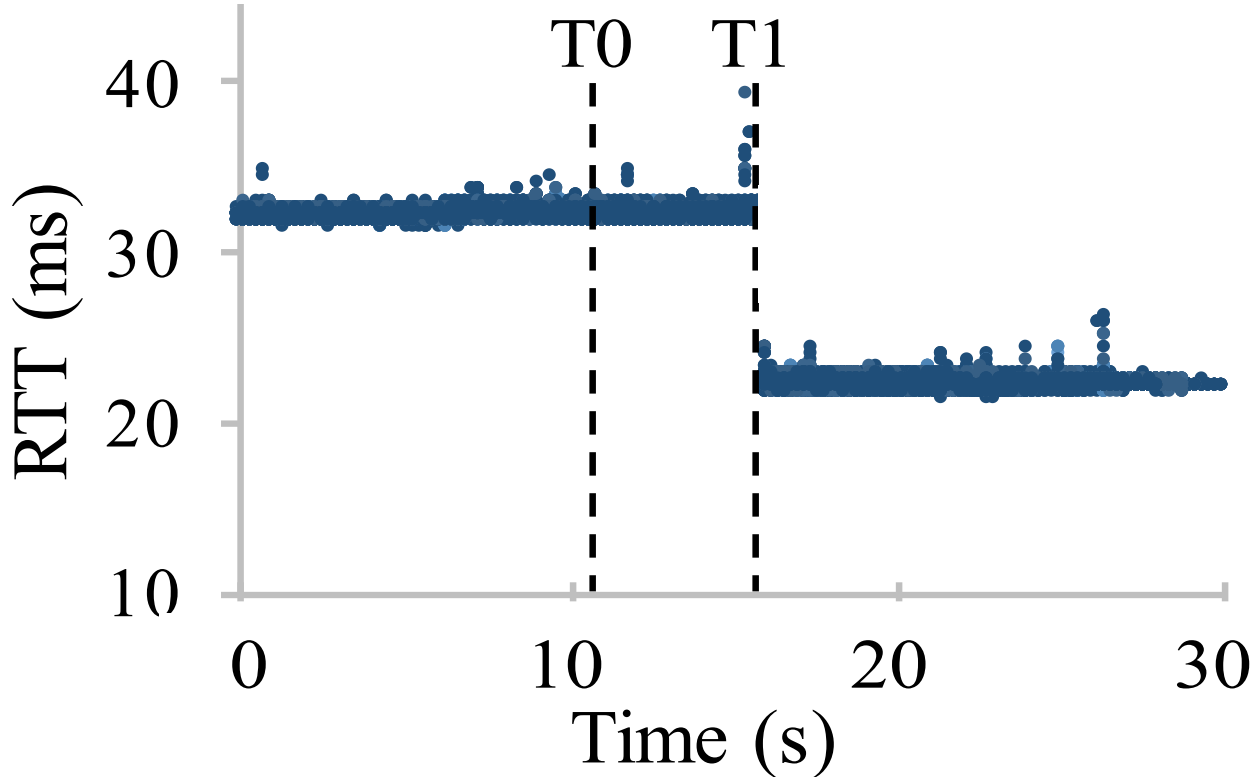
Evaluations

- Service evolution and parallel instances
  - LTE/EPC
  - EPC evolved with SMORE
  - EPC evolved with MobiScud
- Data-centric service management
  - Handling bursts of UE attachments
  - Migrating to a new location
- Scalability
  - End-to-end orchestration request completion time
  - Orchestration time for individual components
  - End-to-end parallel orchestration request completion time

Evaluations

Dynamic PGW state migration

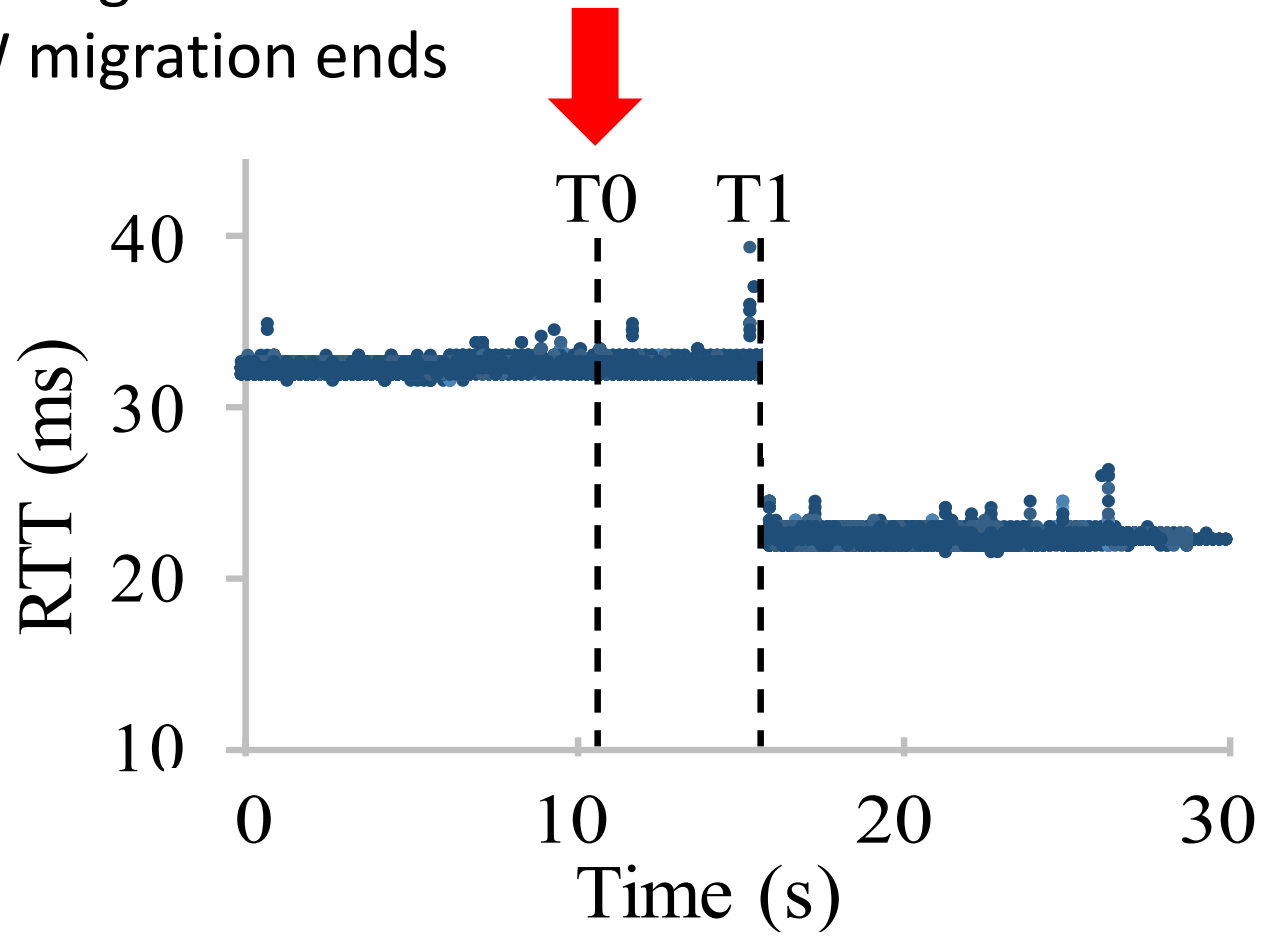
- T0: PGW migration starts
- T1: PGW migration ends



Evaluations

Dynamic PGW state migration

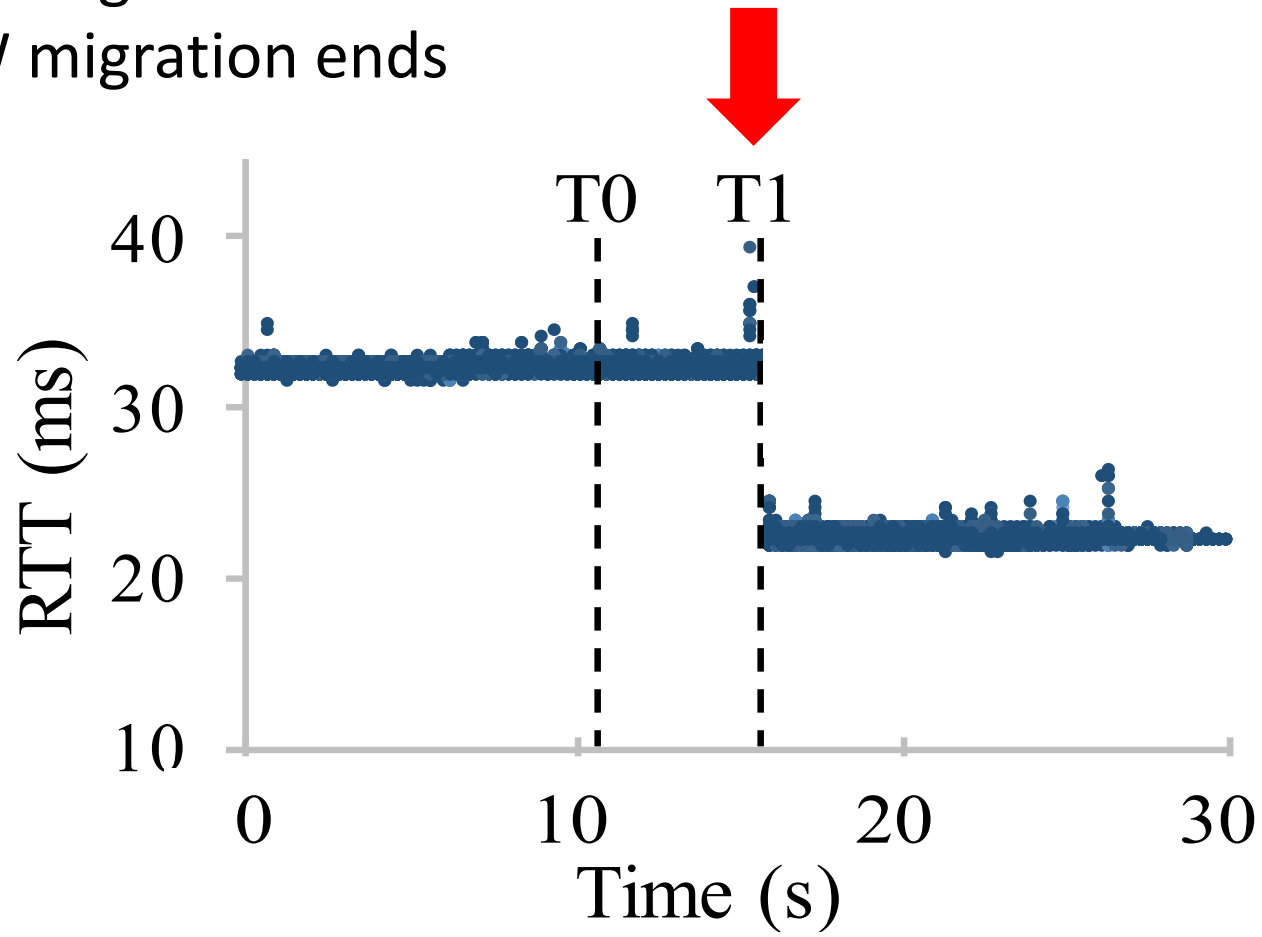
- T0: PGW migration starts
- T1: PGW migration ends

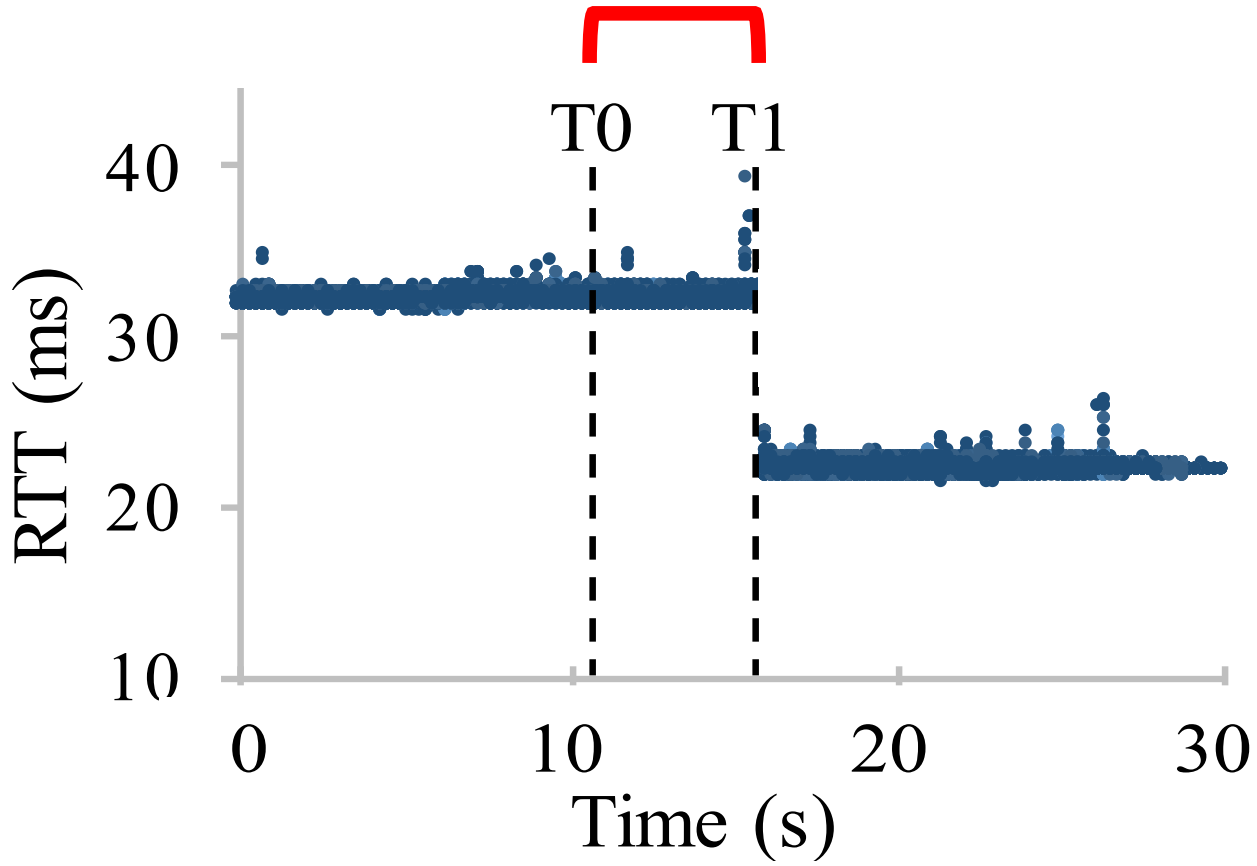


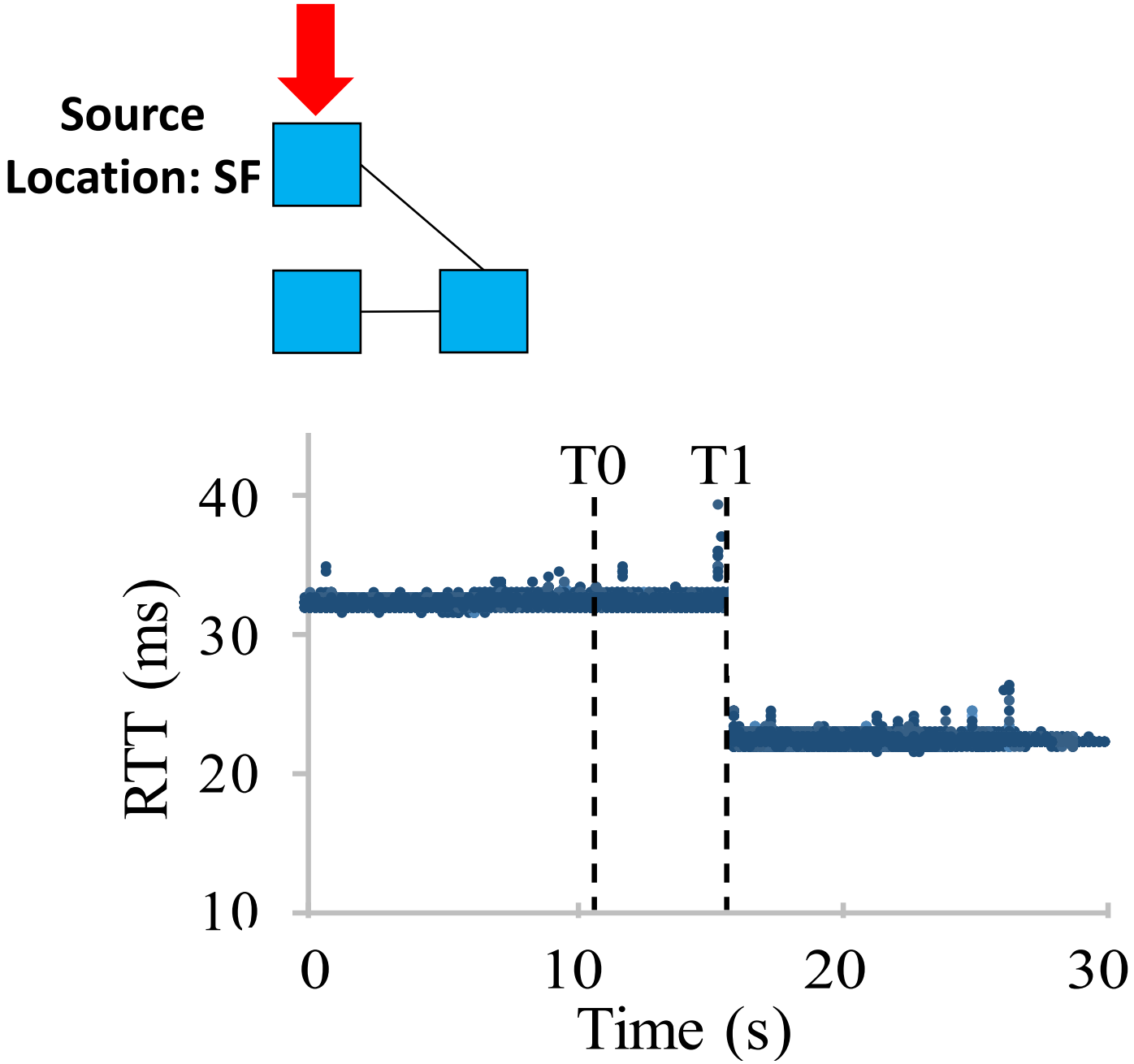
Evaluations

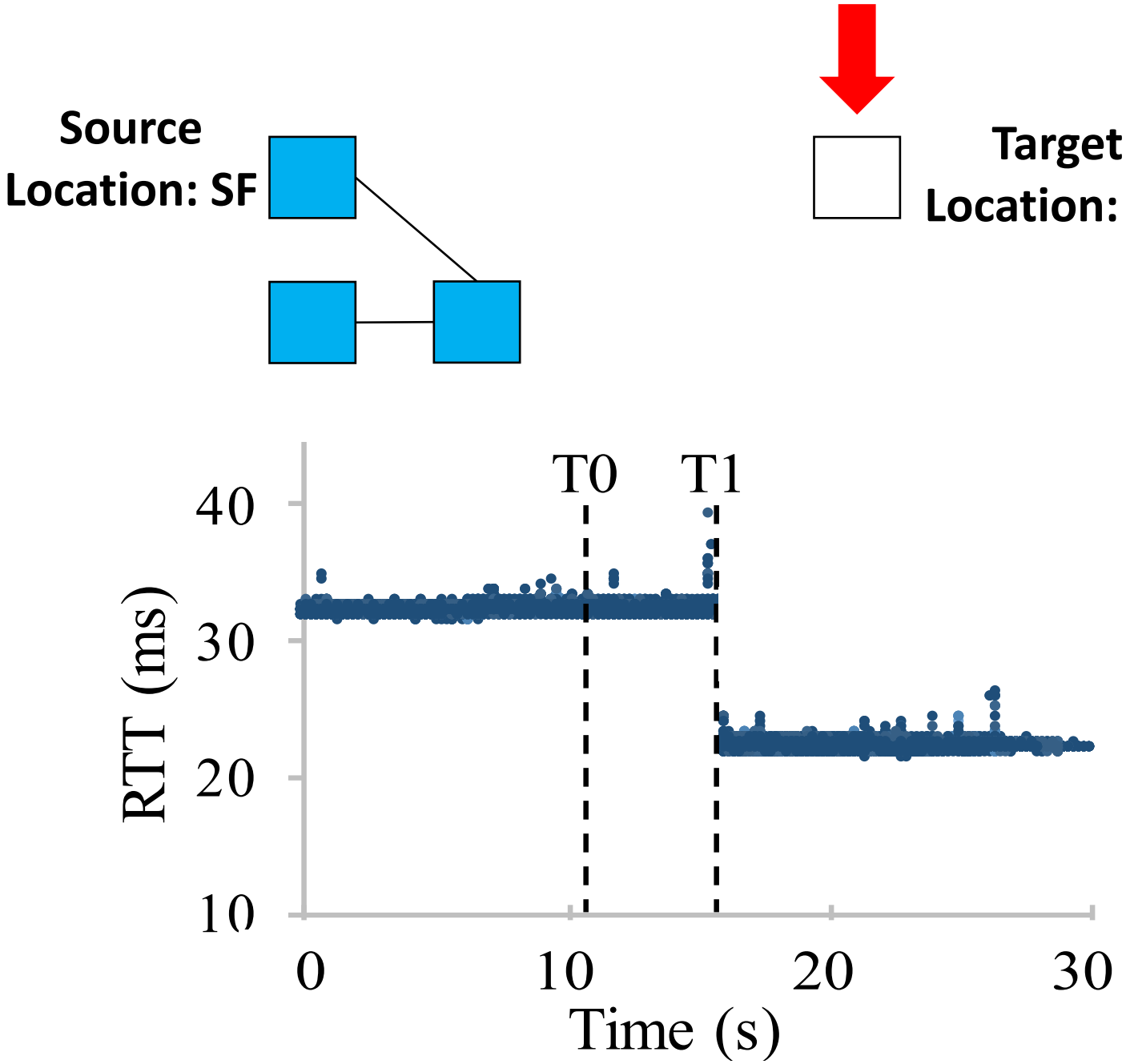
Dynamic PGW state migration

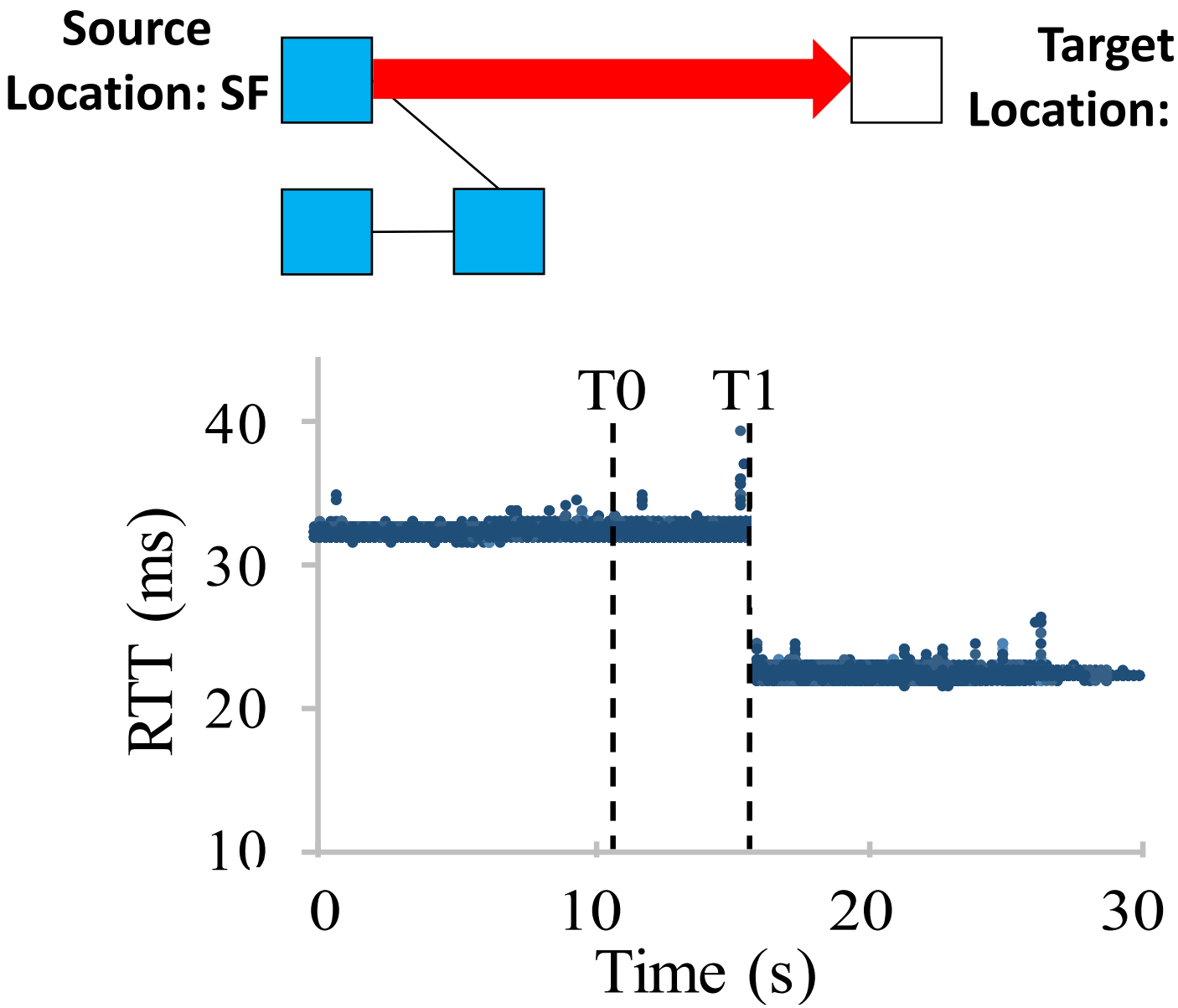
- T0: PGW migration starts
- T1: PGW migration ends

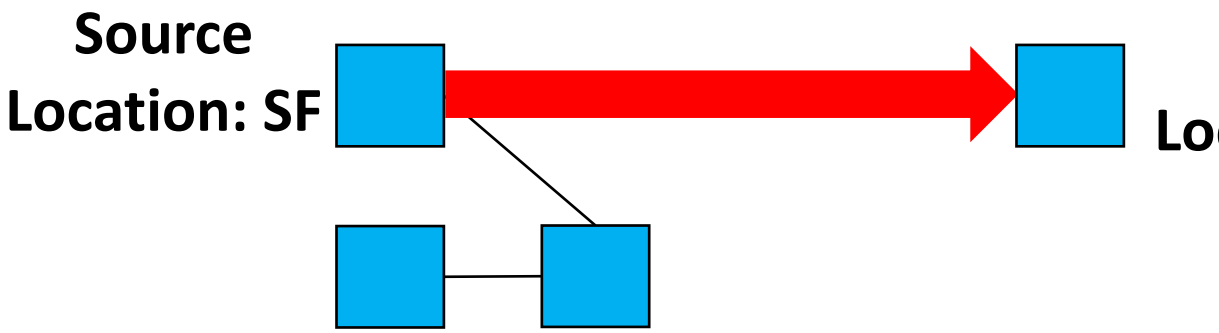


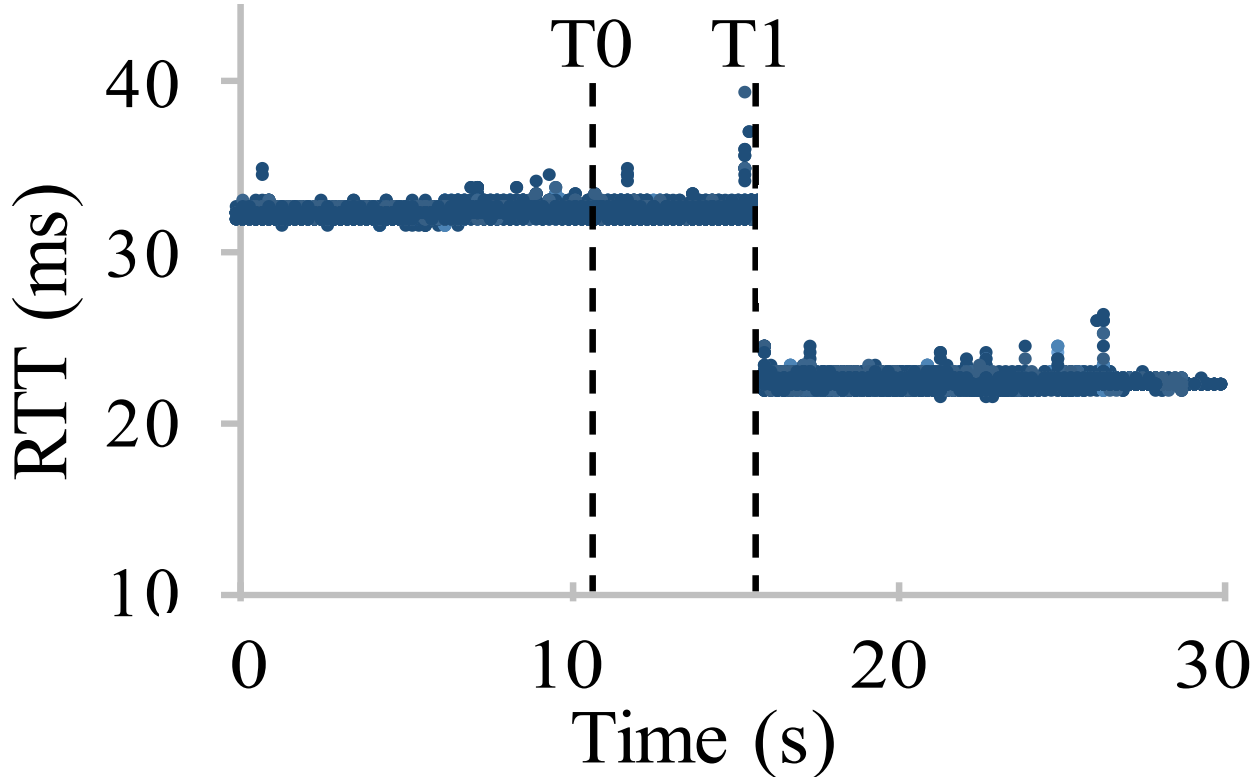

T0
T1
40
30
20
10
RTT (ms)
0
10
20
30
Time (s)

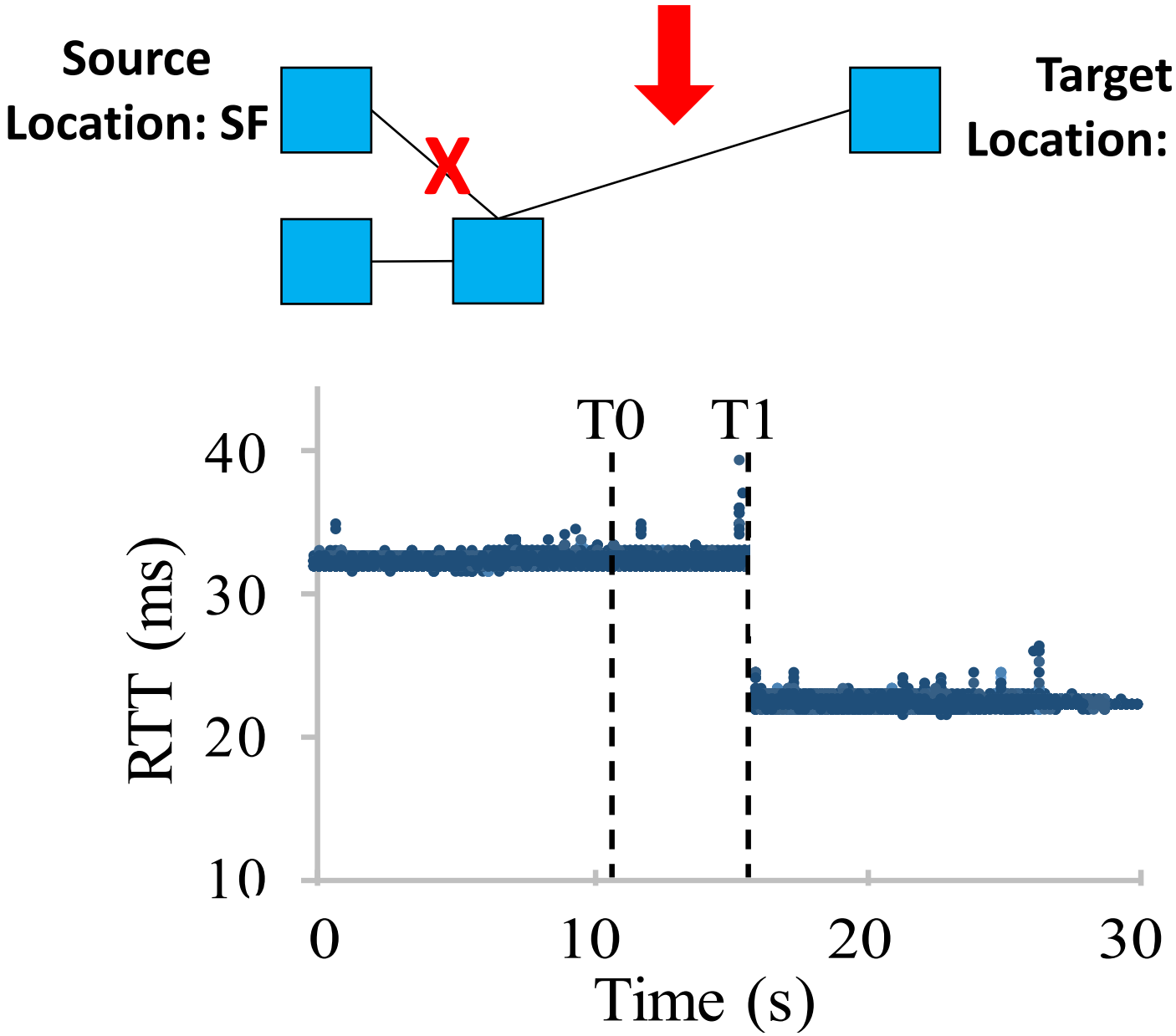

Source
Location: SF
T0
T1
40
30
20
10
RTT (ms)
0
10
20
30
Time (s)

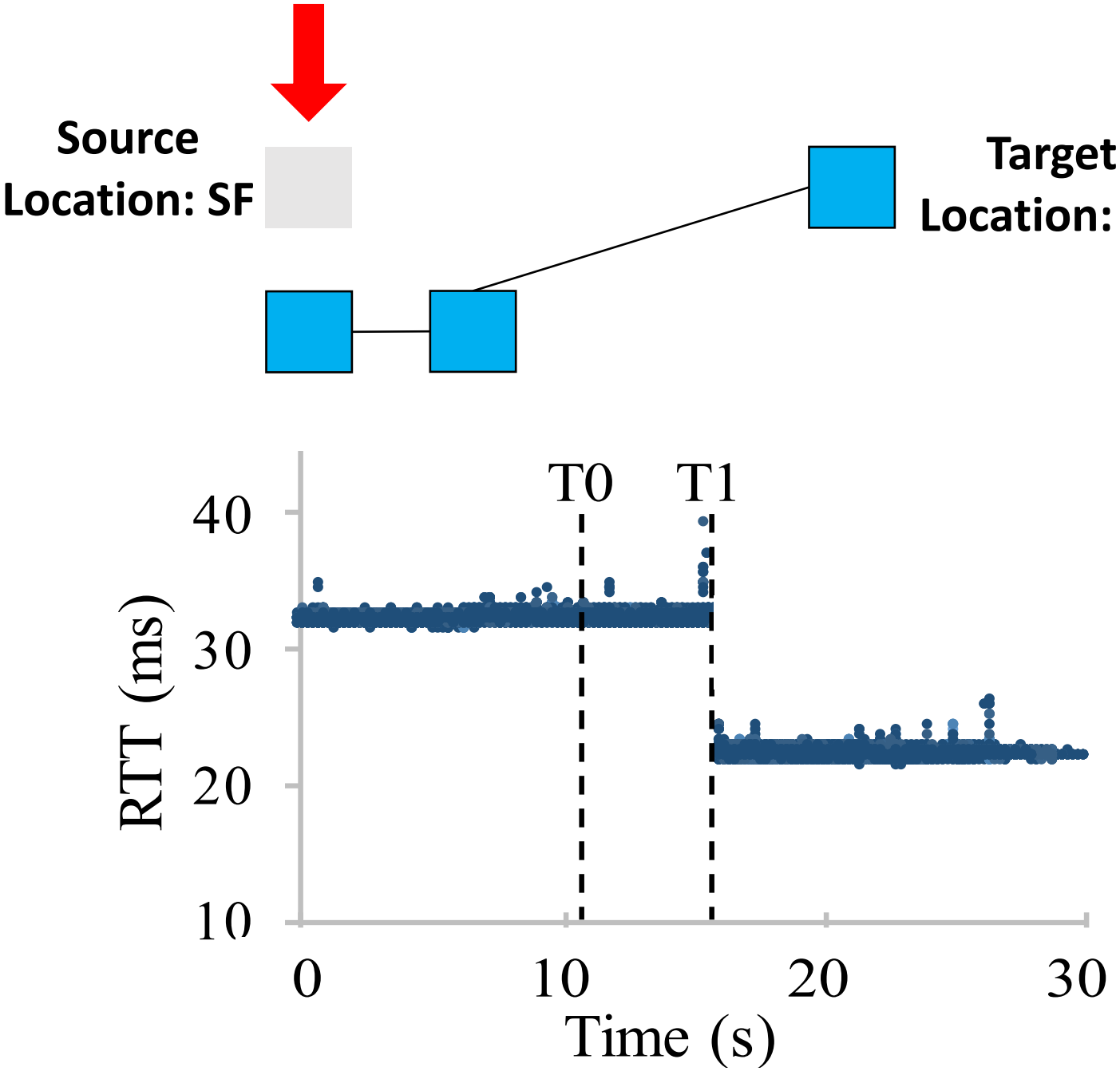

Source
Location: SF
Target
Location: NY
T0
T1
40
30
20
10
RTT (ms)
0
10
20
30
Time (s)


Source
Location: SF
Target
Location: NY
T0
T1
40
30
20
10
RTT (ms)
0
10
20
30
Time (s)


Source
Location: SF
Target
Location: NY


T0
T1
40
30
20
10
RTT (ms)
0
10
20
30
Time (s)


Source
Location: SF
Target
Location: NY
T0
T1
40
30
20
10
RTT (ms)
0
10
20
30
Time (s)


Source
Location: SF
Target
Location: NY
T0
T1
40
30
20
10
RTT (ms)
0
10
20
30
Time (s)

Conclusion

- Problem
  - Mobile network service deployment and evolution very slow
  - Future mobile networks will be unable to meet app demands without being more evolvable
- Our solution
  - Trend towards mobile software-defined infrastructure (SDI)
  - Proteus, a control platform that allows safe and rapid service creation and evolution in a mobile SDI

Questions?