A Web Service- and ForCES-based Programmable Router Architecture

Evangelos Haleplidis[1], Robert Haas[2], Spyros Denazis[1,3], Odysseas Koufopavlou[1]

[1]University of Patras, ECE Department, Patras, Greece,
[2]IBM Research, Zurich Research Laboratory, Rüschlikon, Switzerland,
[3]Hitachi Sophia Antipolis Lab, France,*Spyros.Denazis@hitachi-eu.com*

Seventh Annual International Working Conference on Active and Programmable Networks
November 21-23 2005

Outline

- The Need for Open APIs
- ForCES
  - Concept
  - Description
- FlexiNET
  - Architecture
  - FlexiNET & ForCES
- ForCEG
  - ForCES Scope
  - Concept
  - ForCEG & FlexiNET
  - Architecture
  - Use Case
- Evaluation – Conclusion – Future Work

The Need for Open APIs

"*a programmable network is distinguished from any other networking environment by the fact that it can be programmed from a minimal set of APIs from which one can ideally compose an infinite spectrum of higher level services*"[1].

[1]Andrew Campbell, Herman De Meer, Michael Kounavis, Kazuho Miki, John Vicente, and Daniel Villela, "A Survey of Programmable Networks", ACM SIGCOMM Computer Communications, 1999.

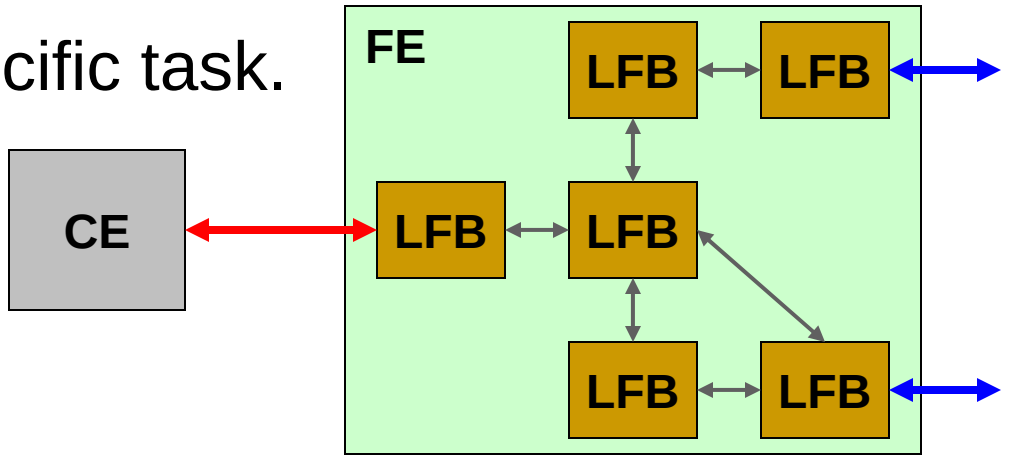
ForCES Concept

- Network division: Forwarding, Control, and Management plane
- Separation:
  - Increased scalability
  - Allows the planes to evolve independently.

- Focus: Communication and Model for separating control-plane functionality (e.g. routing protocols) from data-forwarding-plane per-packet activities (e.g. packet forwarding).

ForCES Description

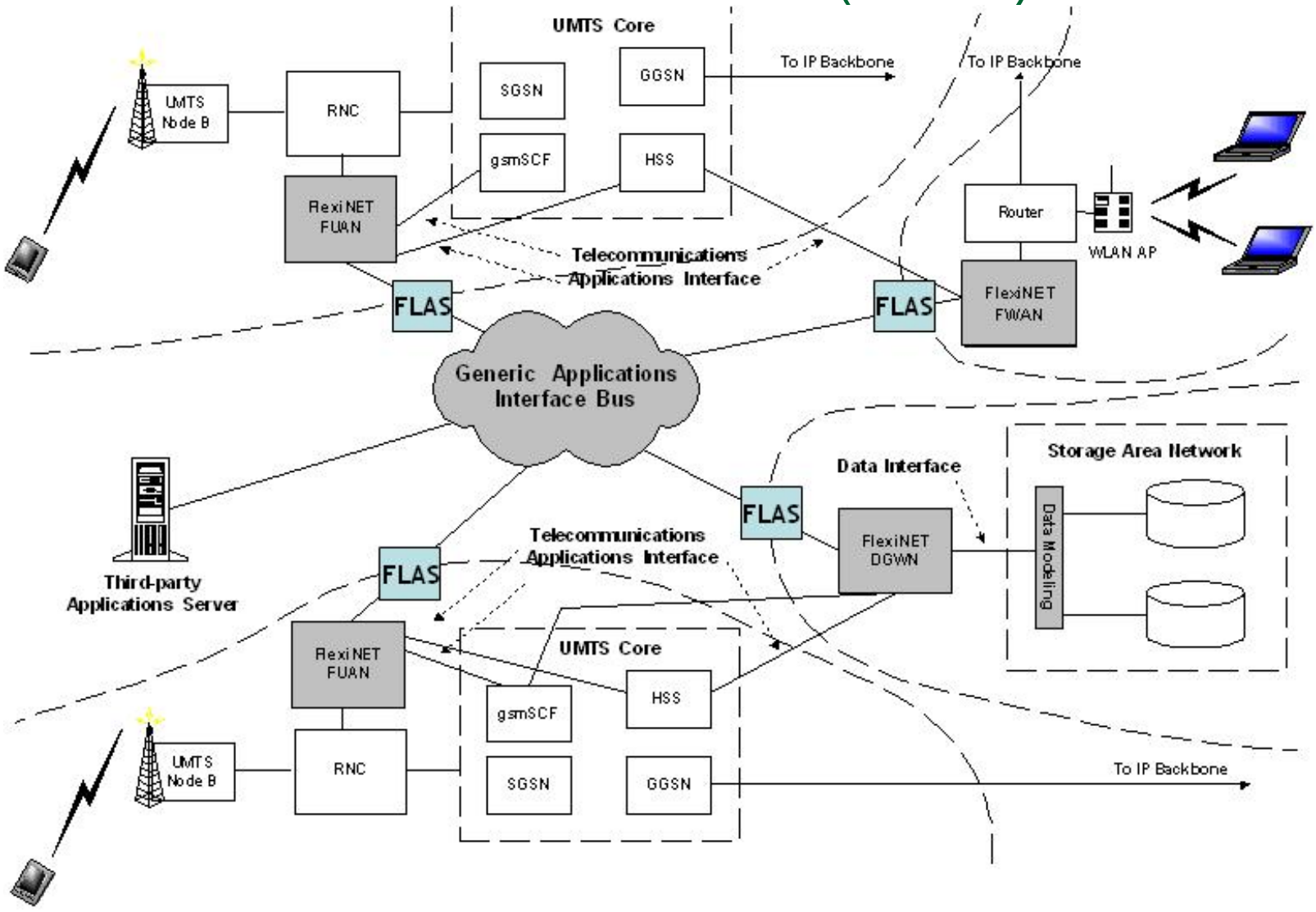
- Control Elements (CEs).
- Forwarding Elements (FEs).
- Logical Function Blocks (LFBs).
  - Encapsulating fine-grained, forwarding plane, operations
  - Responsible of a specific task.
  - Static Topology
  - Dynamic Topology



FlexiNET Architecture

- Define and implement a scalable and modular network architecture incorporating adequate network elements offering cross-connect control, switching/routing control, and advanced services management/access functions at the network access points that currently only support connectivity between user terminals and network core infrastructures.
- FlexiNET Node Instances:
  - FUAN
  - FWAN
  - DGWN
  - FLAS

FlexiNET Architecture (con.)

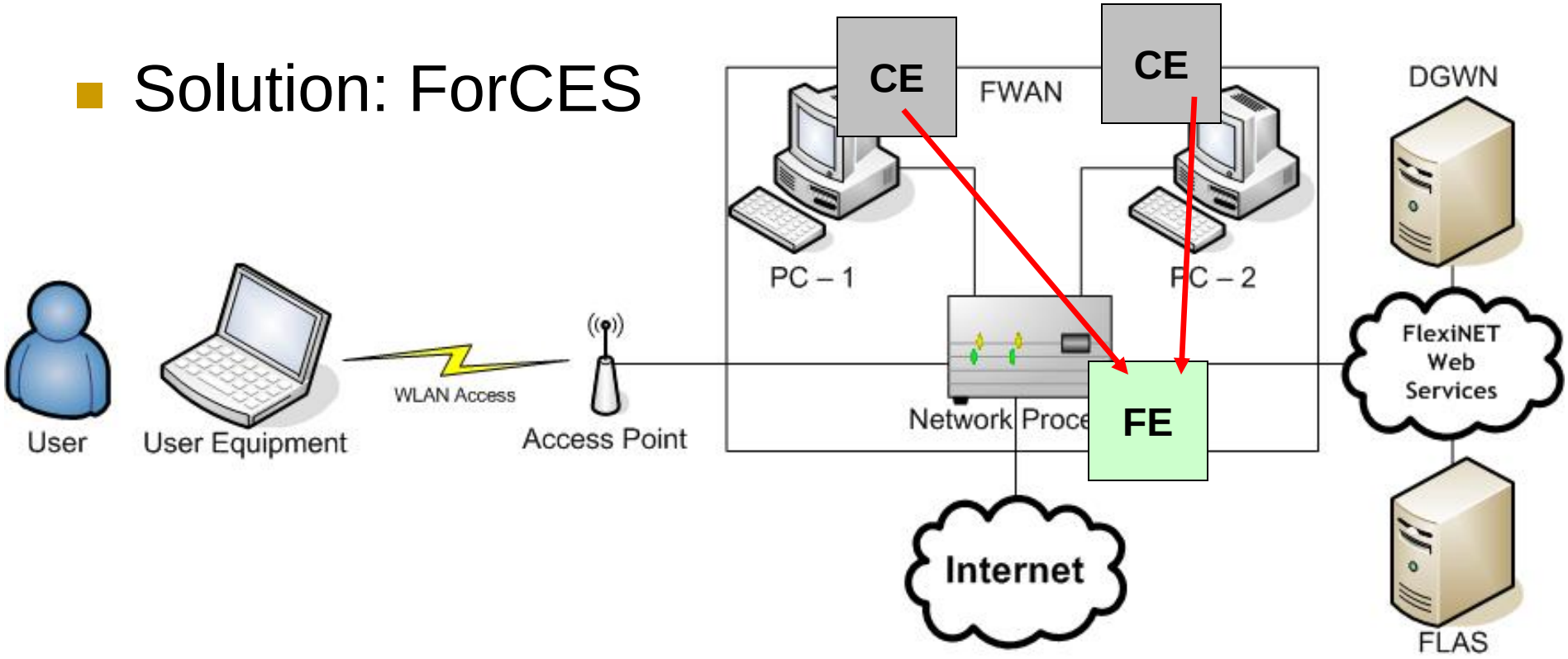


FlexiNET & ForCES

- FlexiNET separates control and data functionalities.
- Dynamically deployed new services require additional programmability functions.
- User specific network customization.

ForCES & FlexiNET

- Dynamically Deploy CEs in any PC.
- User Specific Router Configuration.
- Solution: ForCES



ForCES Scope

- Only modelling of the forwarding plane is within the scope of the ForCES working group.

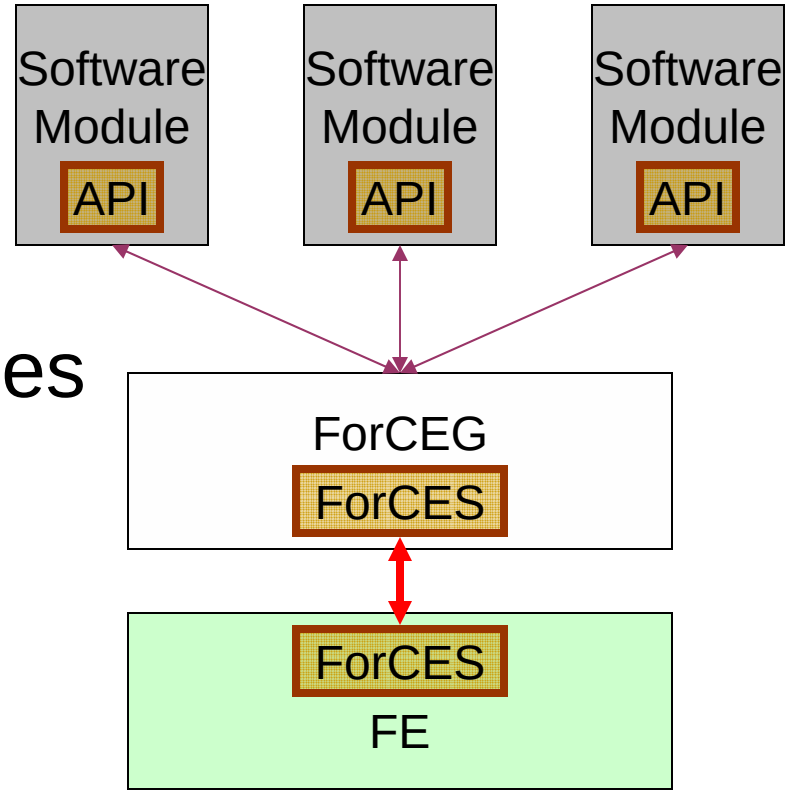
- FEs can only be controlled by a single active CE, hence different services that require state changes in LFBs belonging to the same FE have to go over the same CE.

ForCEG Concept

- Further separate the functionalities of the control point into the Main Control Programs (MCPs), which execute control-plane services, and a ForCES CE Gateway (ForCEG), which implements, among other things, the CE side of the ForCES protocol

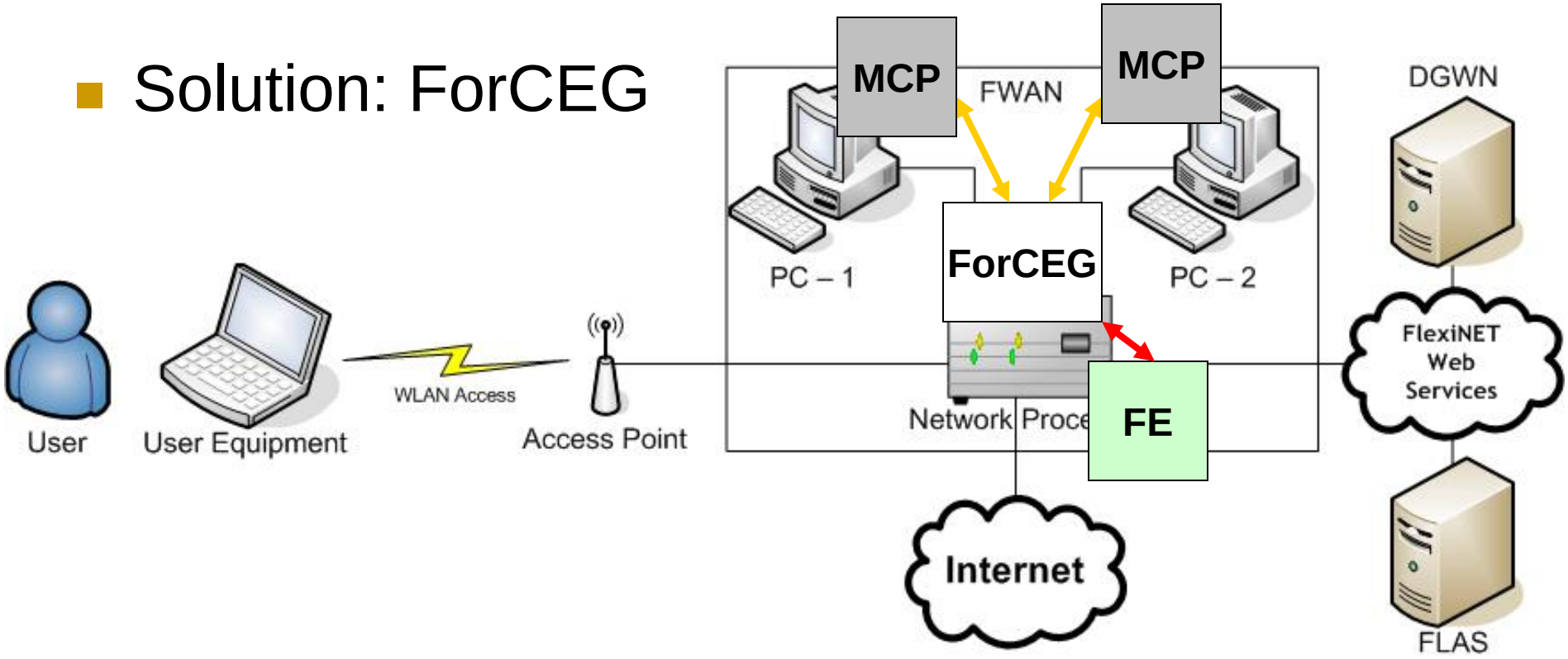
ForCEG Concept (con.)

- Conceal ForCEs Model Details
- Multiple MCPs to FEs
- FE Easy Discovery
- Generic API
- Solution: Web Services

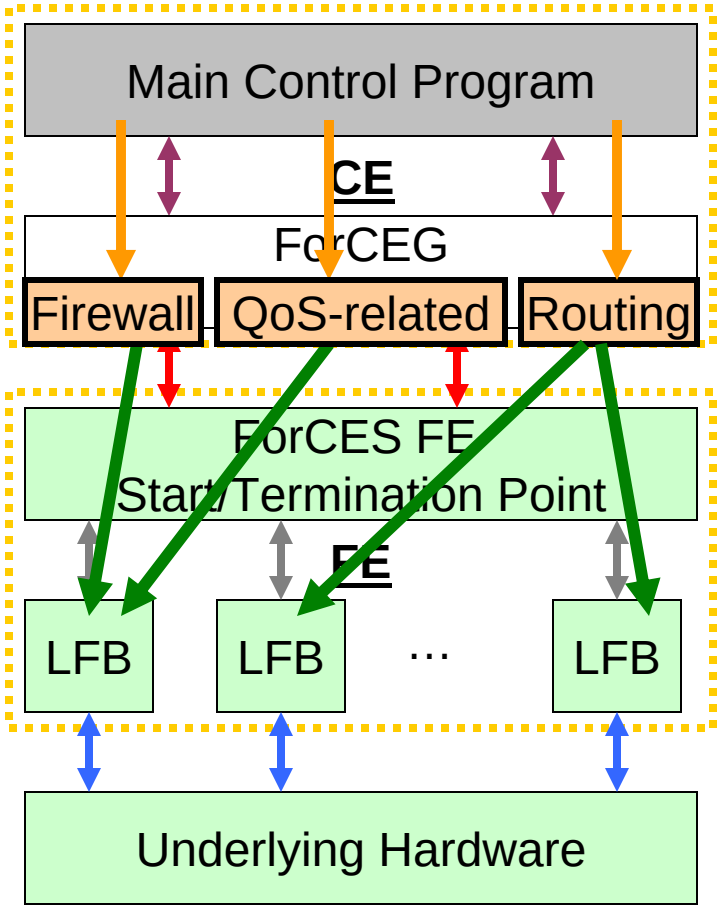


ForCEG & FlexiNET

- Dynamically Deploy MCPs in any PC.
- User Specific Router Configuration.
- Solution: ForCEG



ForCEG Concept (con.)



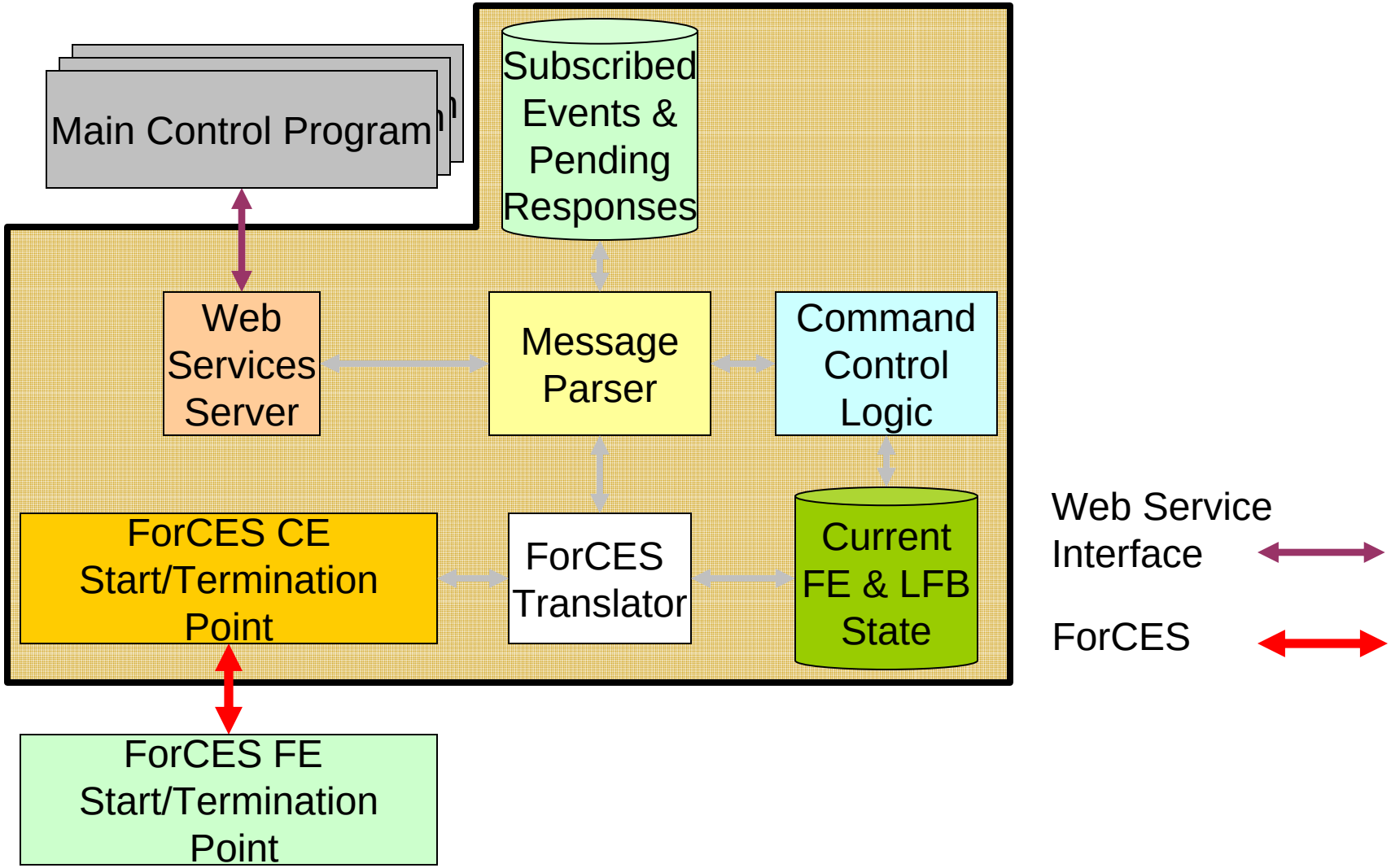
Target

Correlation between high-layer function & LFB's

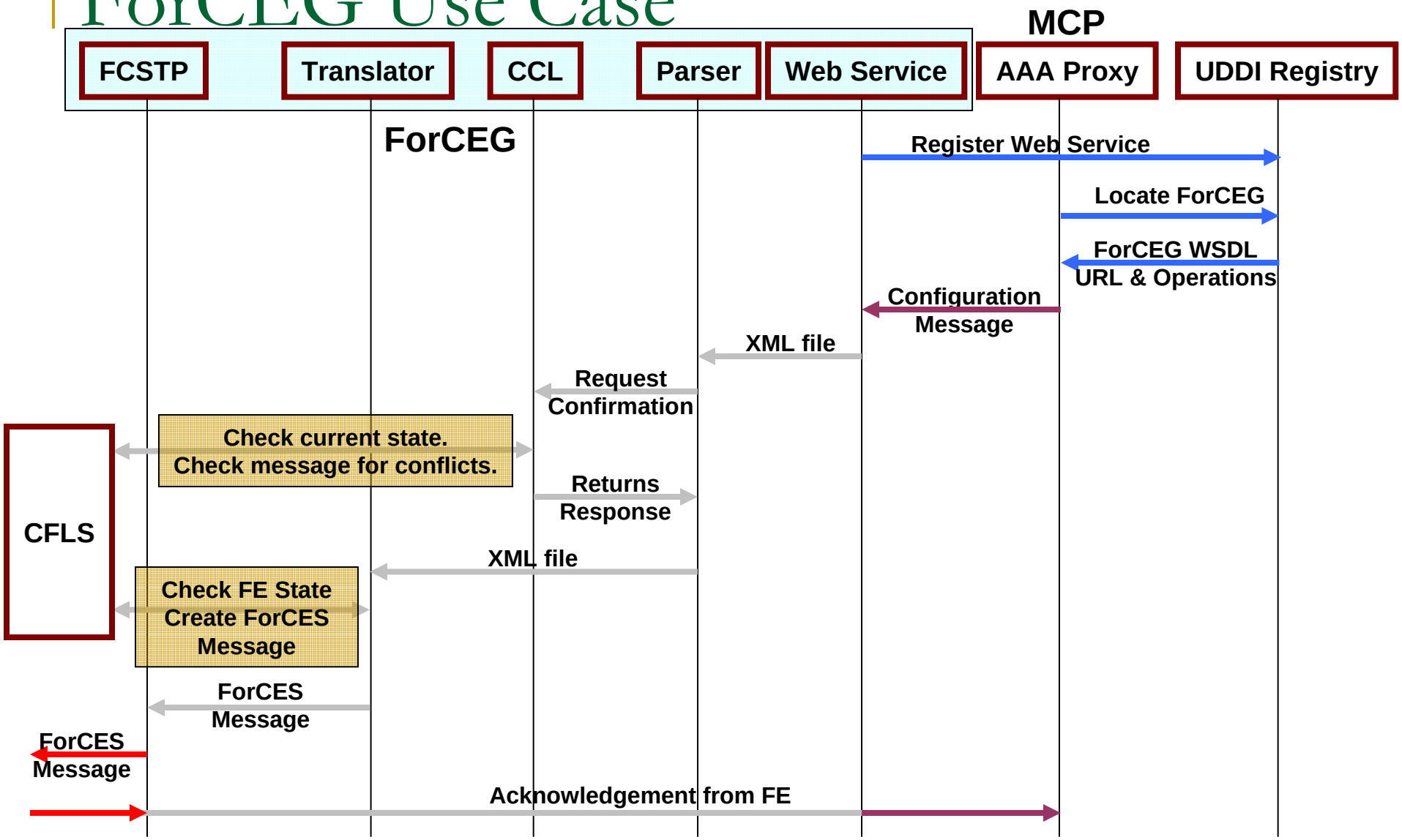
ForCEG Concept (con.)

- Translates configuration commands from a Generic Web Service API into ForCES packets.
- Extend ForCES protocol.
- Conceal ForCES model from high-level functions.
- Provide connections of multiple Contol Elements into a Forwarding Element.
- Advertise APIs through a UDDI registry.
- Detect and prevent conflicts between different MCPs.

ForCEG Architecture



ForCEG Use Case



Evaluation

- ForCEG will be evaluated against:
  - Performance.
  - Versatility.
  - Ease of use.
- Measurements:
  - Delay between the Web Service Call and the sending of the ForCES packet.
  - Overhead incurred by the architecture as a service executes.

Conclusion

- Motivation: To create a Generic Web Service API to provide access to ForCES APIs.
- Heterogeneous: Need a middleware architecture approach which provides a Generic Service API and translates it into a low-level API.
- Contribution: Our approach extends the ForCES protocol and addresses the issues of multiple CEs to FE.

Future Work

- Dynamic Mapping in an automated way.
- Integration of other control protocols such as Netconf may extend the versatility of the ForCEG.
- Dynamically addition of user specific mappings.

Questions?