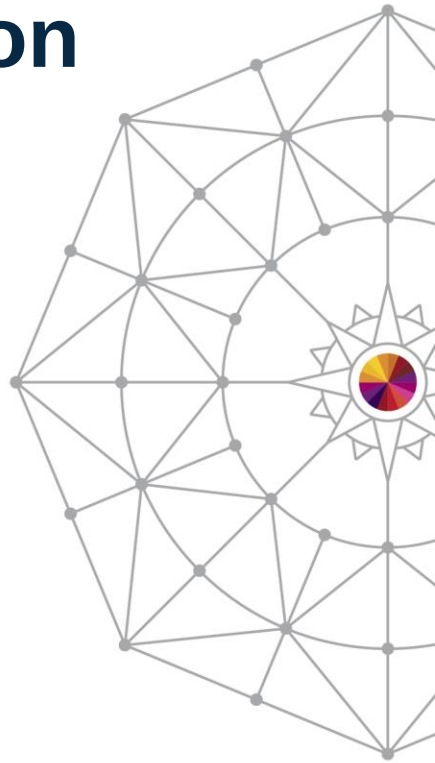
Leveraging BCPii in Automation

Zachary Williams
CA Technologies

August 4, 2014
Session #16090



Agenda

- BCPii
- Use Case
- Discussion

Goal

How do we make our jobs easier by using BCPii in automation.

- Glossary
  - BCPii = Base Control Program internal interface
  - SE = Support Element
  - OOCOD = On/Off Capacity On Demand
  - CAPREC = Capacity Record
  - CF = Coupling Facility
  - MSU = Million Service Units
  - (E)CSA = (Extended) Common Services Area
  - MQ = Message Queuing (IBM WebSphere MQ)
  - CPC = Central Processor Complex
  - HMC = Hardware Management Console
  - CBU = Capacity Back Up
  - ZBX = system Z Blade center eXtension
  - GP = General Processor
  - IFL = Integrated Facility for Linux
  - zIIP = z/OS Integrated Information Processor
  - WLM = Work Load Manager

BCPii - Overview

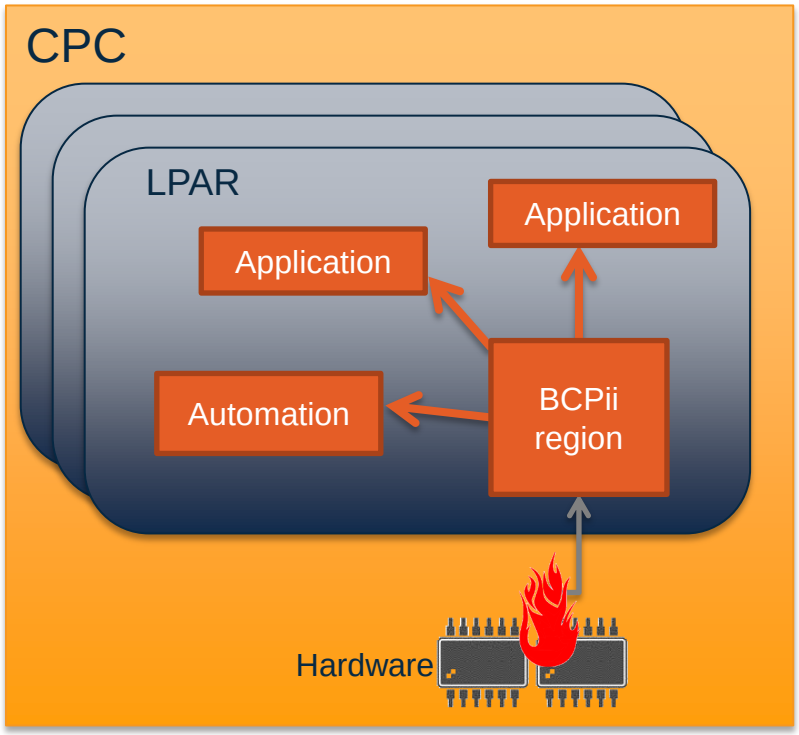
Events Attributes Commands

Programmatically access HMC hardware functionality.

BCPii - Overview

Events Attributes Commands

Applications can register for hardware and software events on the current CPC.



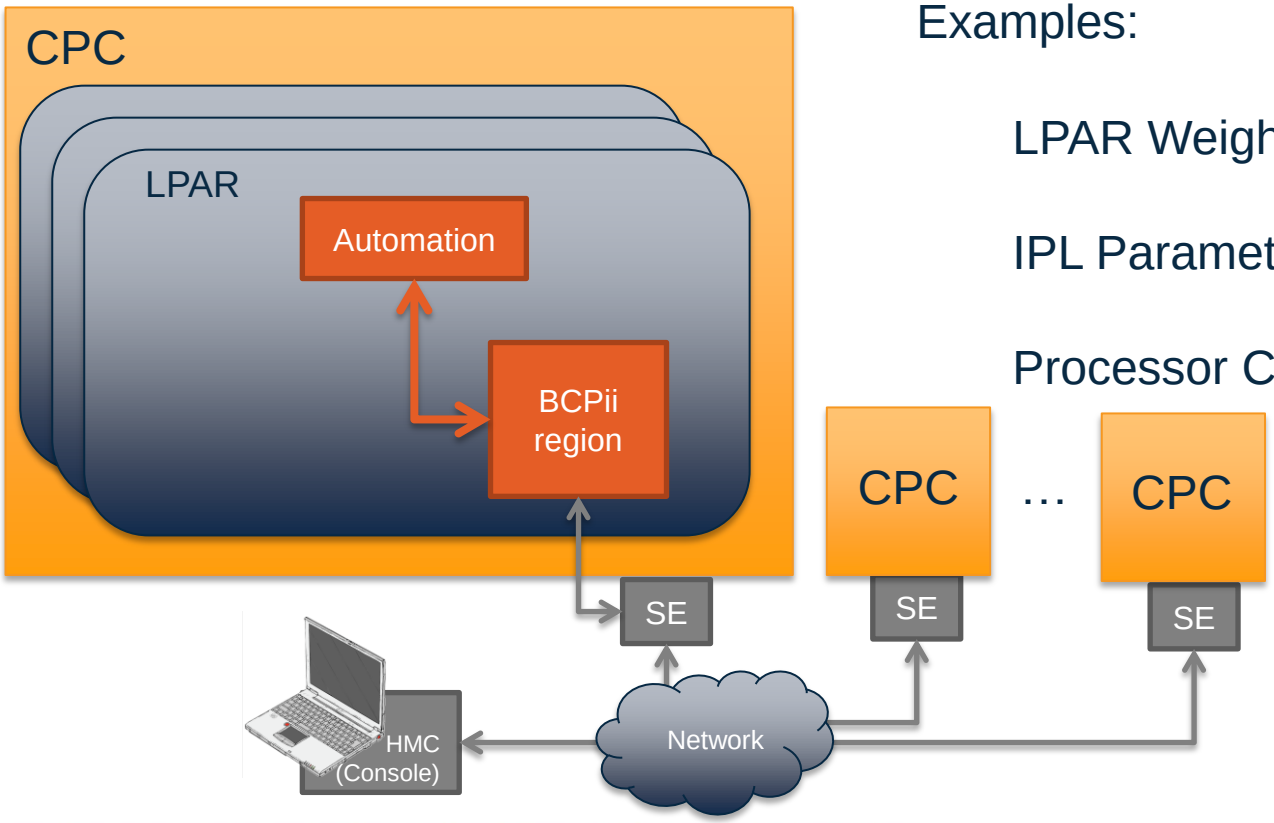
Examples:

- Activation Profile Change
- Capacity Record Change
- Hardware Message
- Command Response

BCPii - Overview

Events Attributes Commands

Applications can securely and programmatically retrieve or update HMC managed objects associated with CPCs, LPARs and Activation Profiles.



Examples:

LPAR Weights,

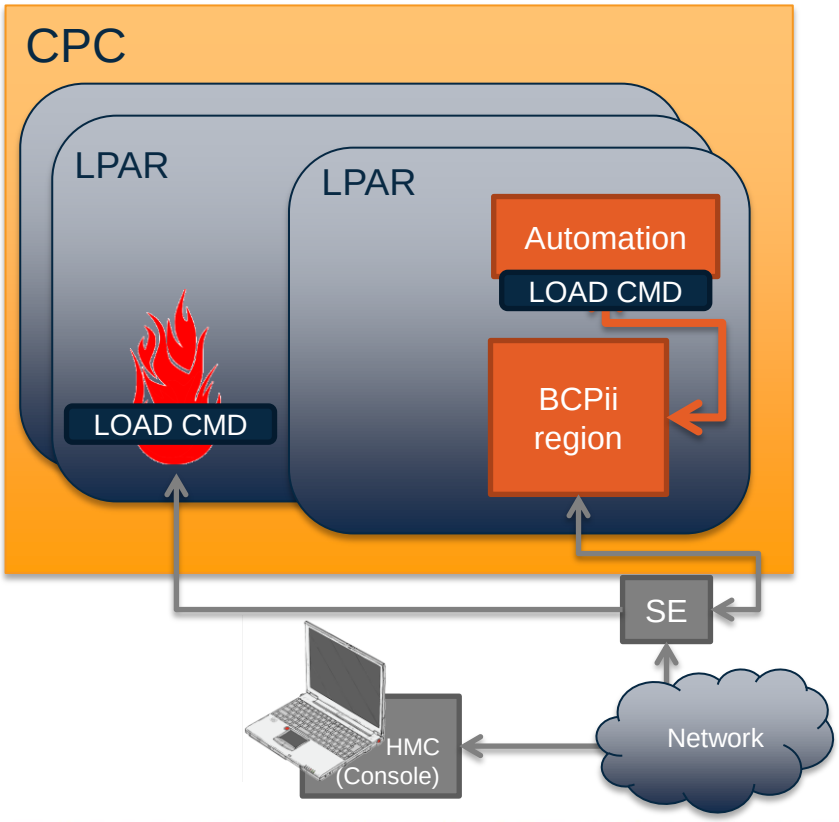
IPL Parameters

Processor Configuration

BCPii - Overview

Events Attributes **Commands**

Applications can issue commands asynchronously against HMC managed objects.



Examples:

Capacity (OOCOD/CBU/tempcap)

Activate/Deactivate/Load

Power Control

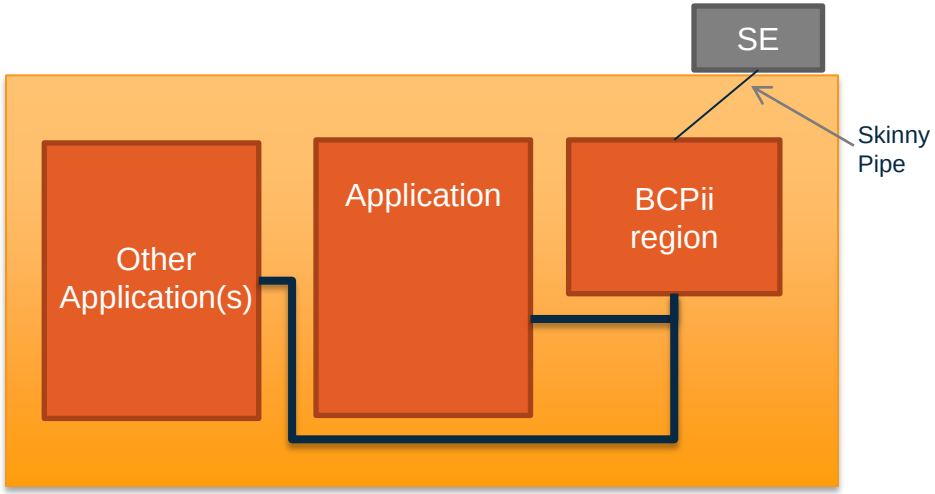
BCPii - Tips, tricks, and other thoughts

Caching	Enhanced Topology	Deriving Attributes	Security

BCPii - Tips, tricks, and other thoughts

Caching | Enhanced Topology | Deriving Attributes | Security

Issue: BCPii is hardware constrained and retrieval time of certain attributes can be slow. Also many attributes do not change frequently.

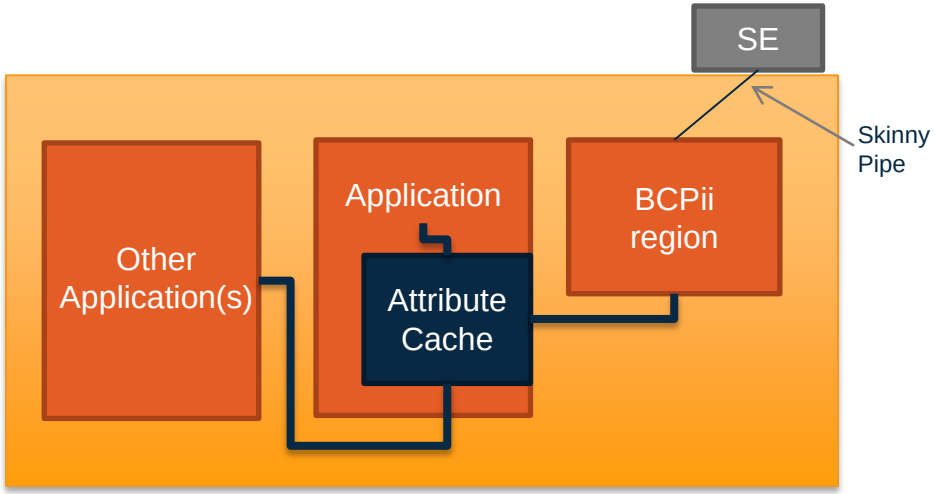


BCPii - Tips, tricks, and other thoughts

Caching | Enhanced Topology | Deriving Attributes | Security

Issue: BCPii is hardware constrained and retrieval time of certain attributes can be slow. Also many attributes do not change frequently.

Solution: Caching the attributes locally makes access time super fast.

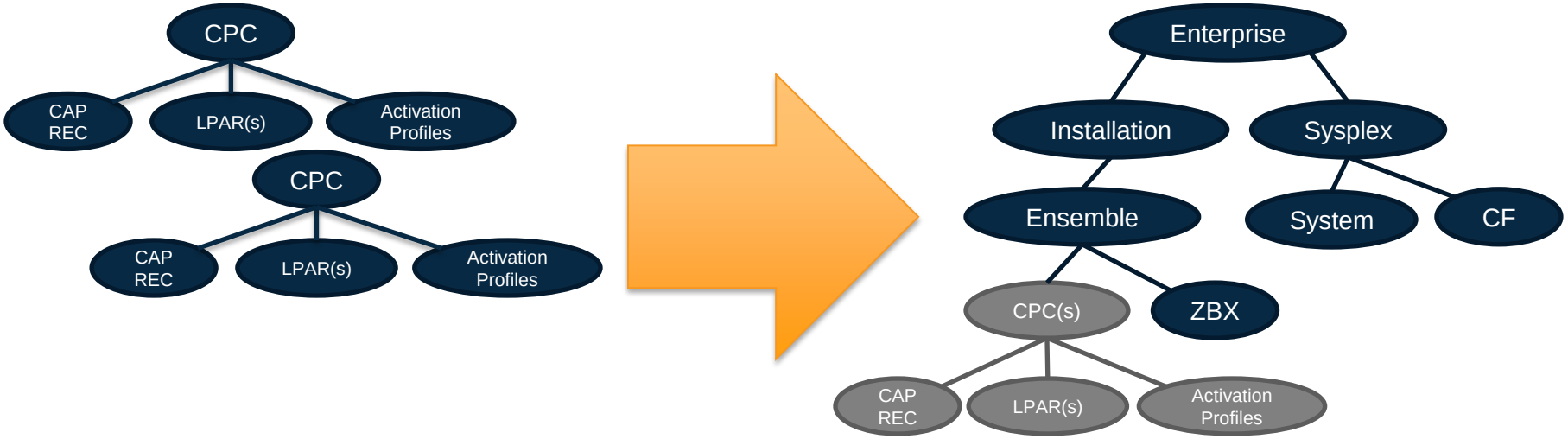


BCPii - Tips, tricks, and other thoughts

Caching | Enhanced Topology | Deriving Attributes | Security

Issue: For our use the BCPii's topological model was restrictive. For example the current model doesn't allow for dynamic addition of CPCs.

Solution: Create an enhanced topology extending the hierarchy of elements.



Increase flexibility and scalability

BCPii - Tips, tricks, and other thoughts

Caching | Enhanced Topology | Deriving Attributes | Security

Issue: The relationship between attribute values and their names can be challenging to understand for software people.

Solution: Create additional attributes which are more consistent and contain more meaningful values; They are derived from original attributes.

```
OPERSTAT  ("OPERATING")
IMLMODE   ('ESA-390TPF', 'CF-PROD')
OPSYS     ('MVS', 'VSE', 'OTHER')
OSLEVEL   (OS dependent release level)
```

```
OPSYSNAME = z/LINUX
OPSYSVERS = aa.bb.cc
```

Treat as normal attributes and cache all of them

BCPii - Tips, tricks, and other thoughts

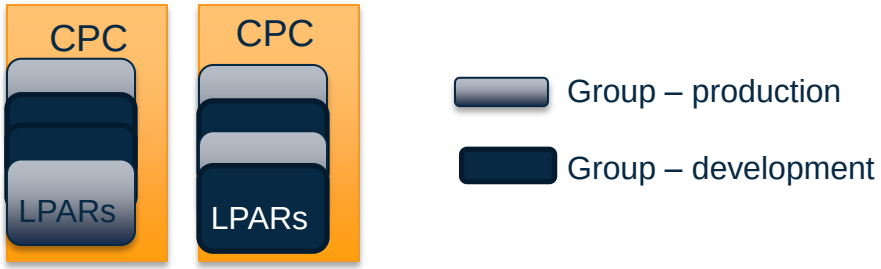
Caching | Enhanced Topology | Deriving Attributes | Security

Issue: We found the BCPii security model to be constraining. For example, when attempting to having different permissions levels for the same type of hardware entity.

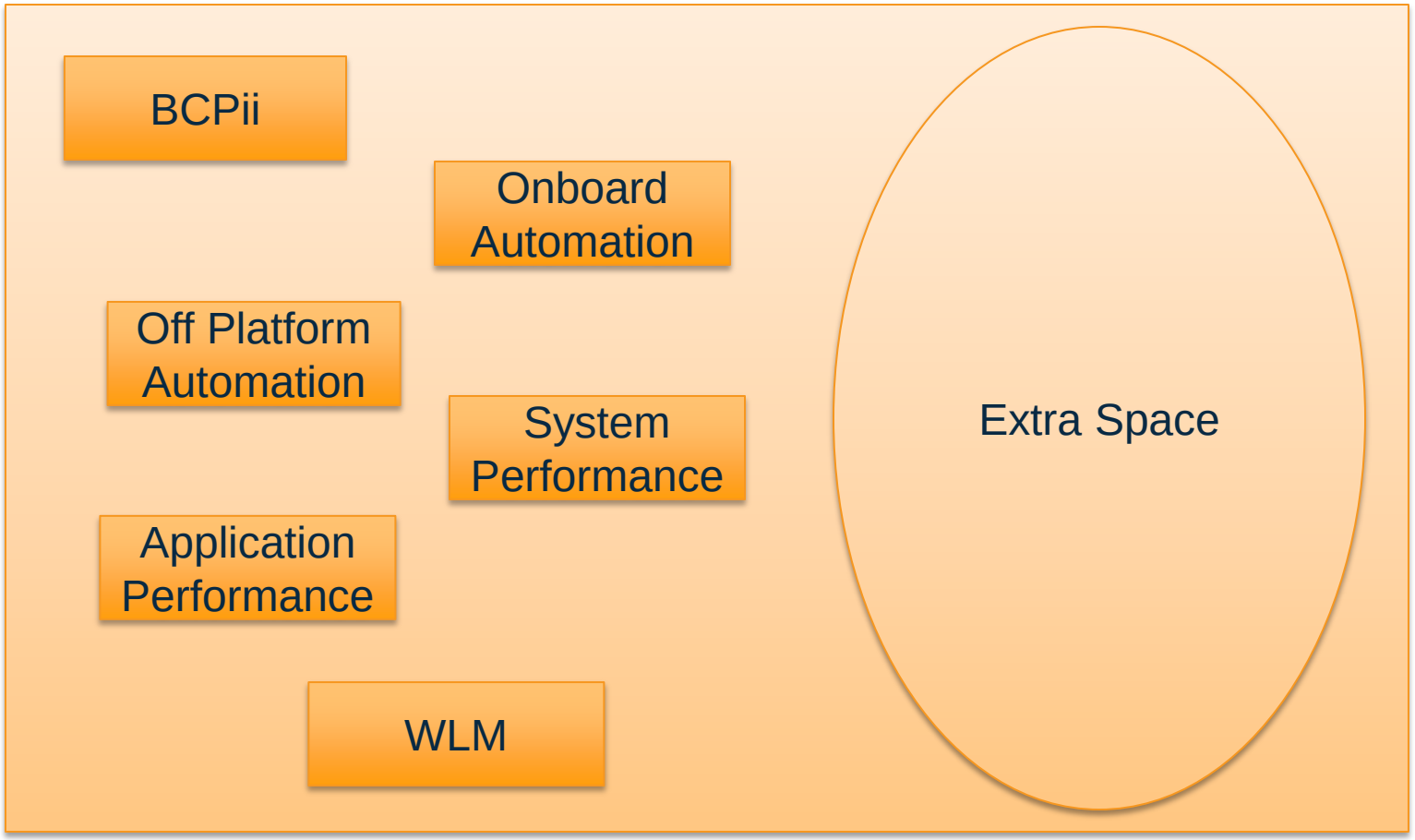
Solution: Create a different security model which allows for user defined groups.

Some Features:

- Allow different types of hardware entities to be grouped together
- Allow/Deny access to a subset of a specific entity type
- Make read only access the default (no security definitions required)



Tools in the Box



Automation

- OPS/MVS / Automation Point / other automation products

On Board

Able to perform automated tasks based on hardware and performance related events.

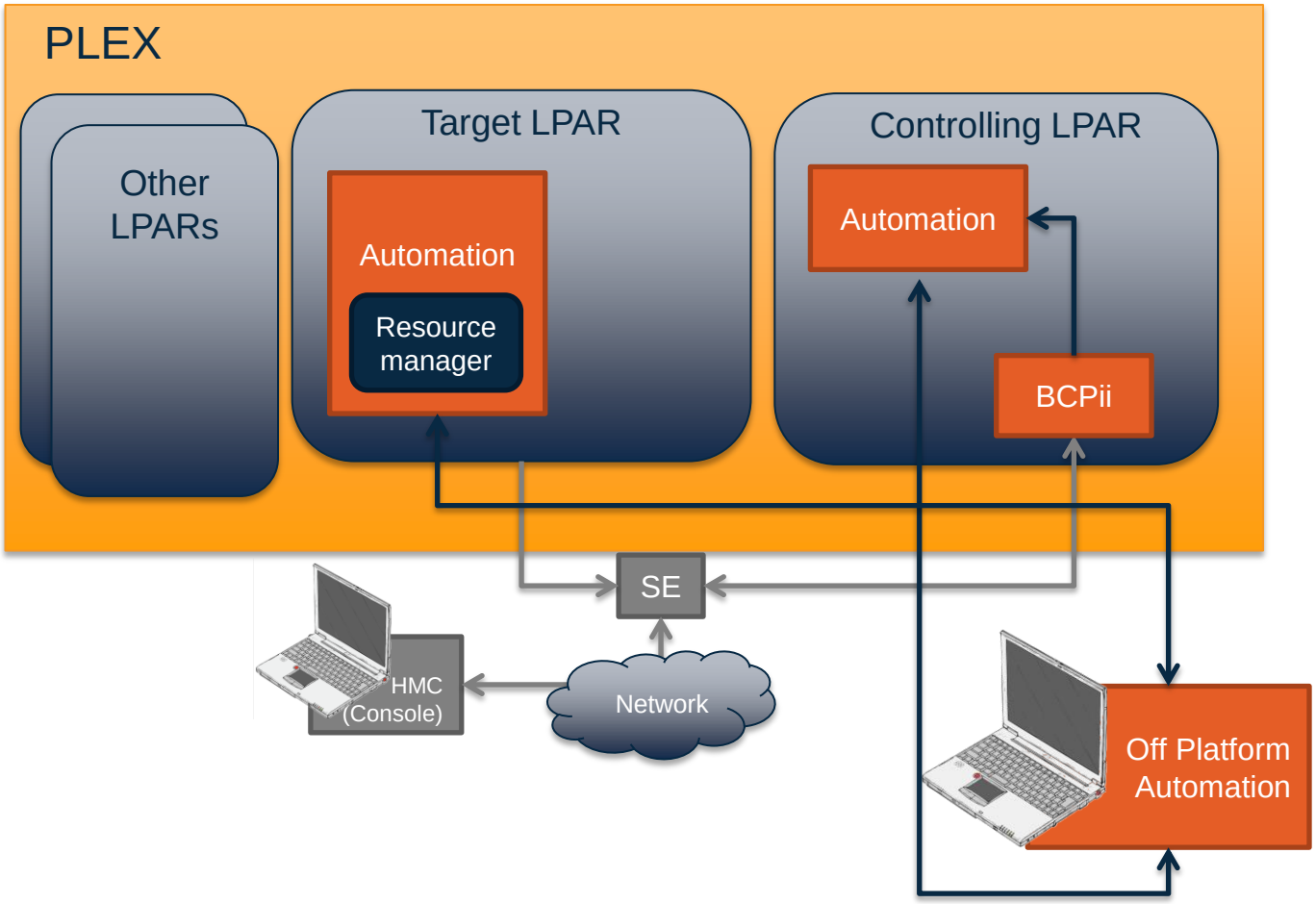
Able to interface with BCPii to automate hardware related actions.

Off platform

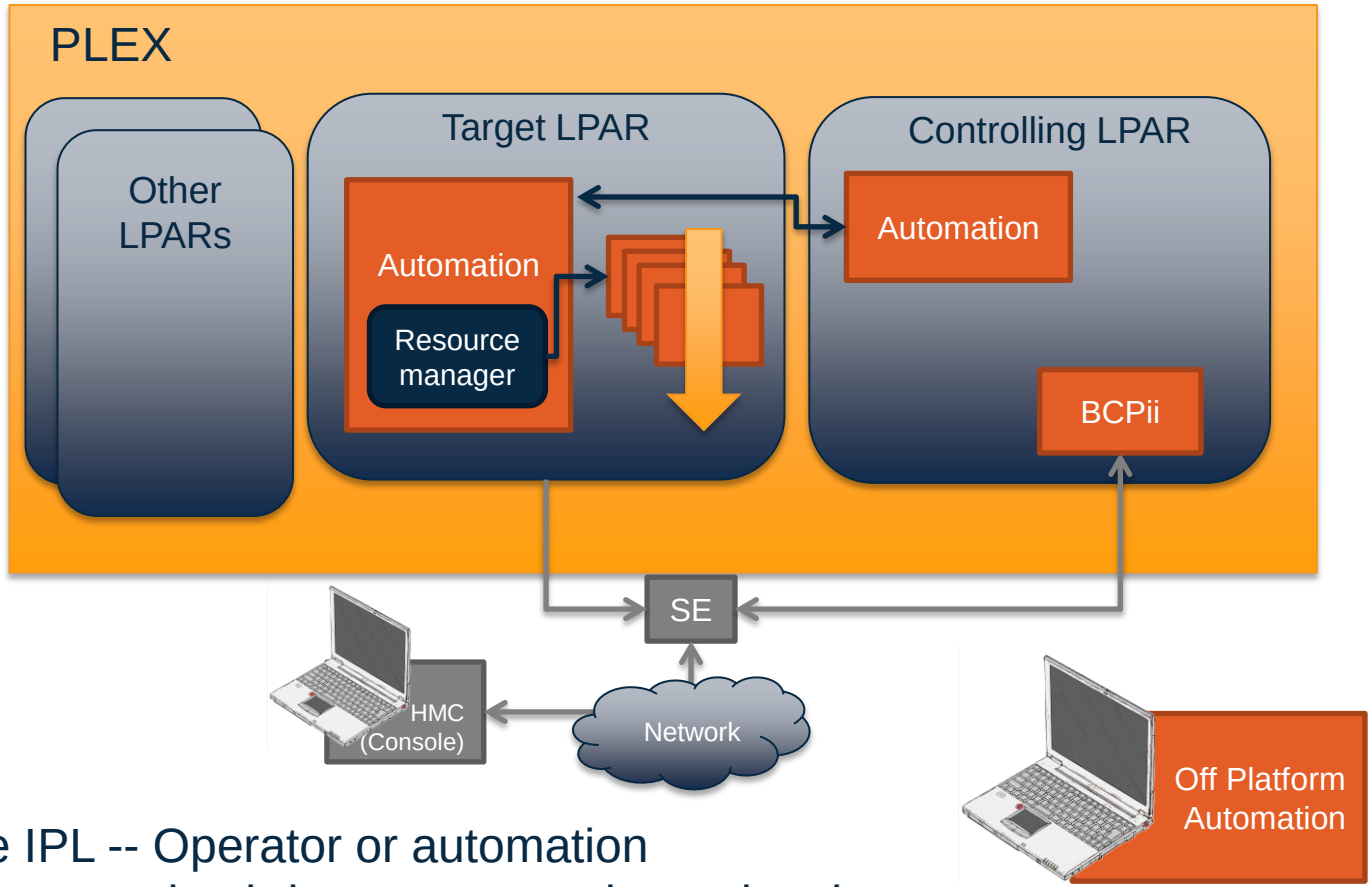
Same hardware interaction capabilities as Onboard Automation

Able to see entire installation from the outside, not trapped in the environment as it is being changed.

Use Case: Automated IPLs

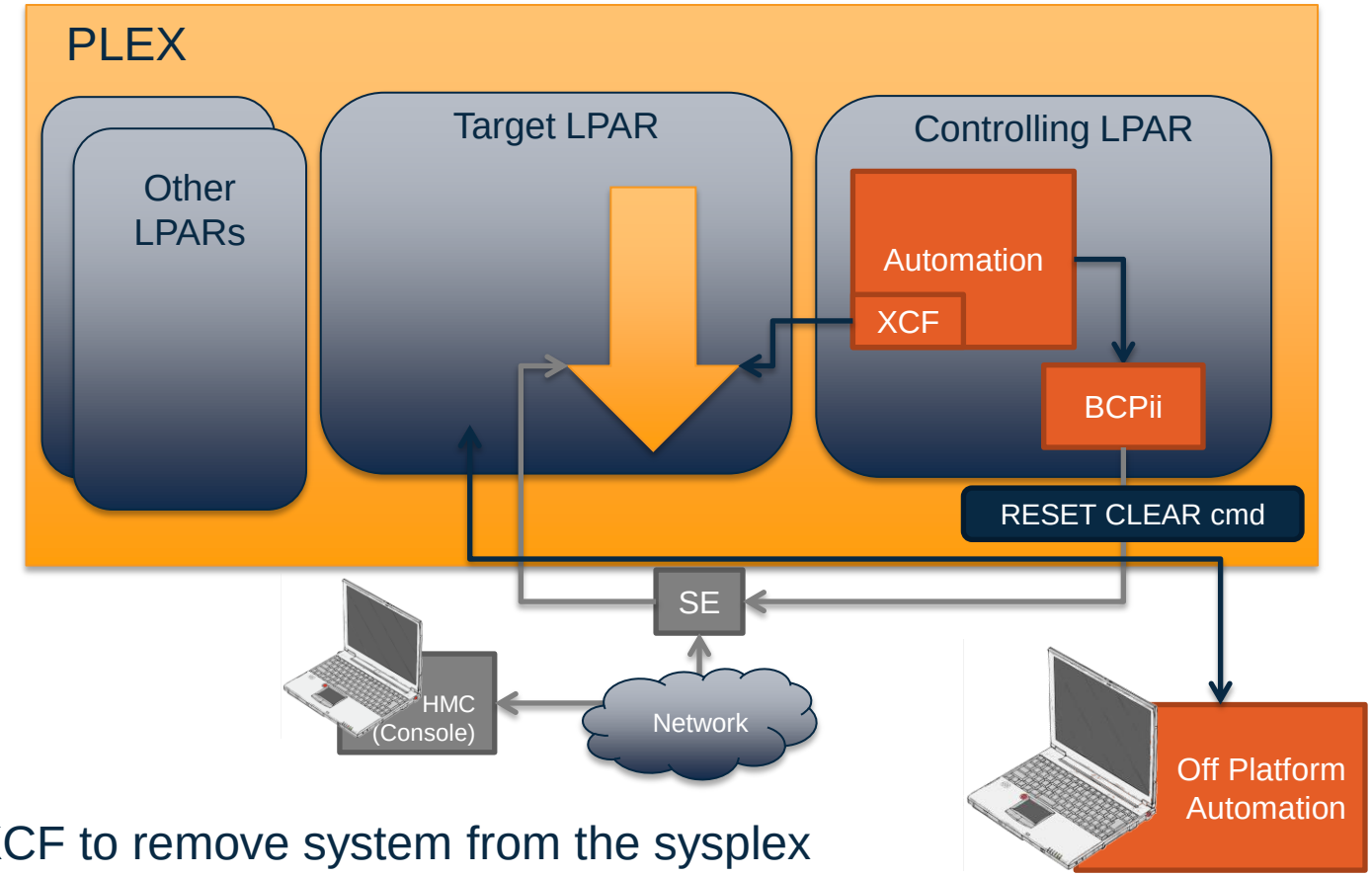


Use Case: Automated IPLs



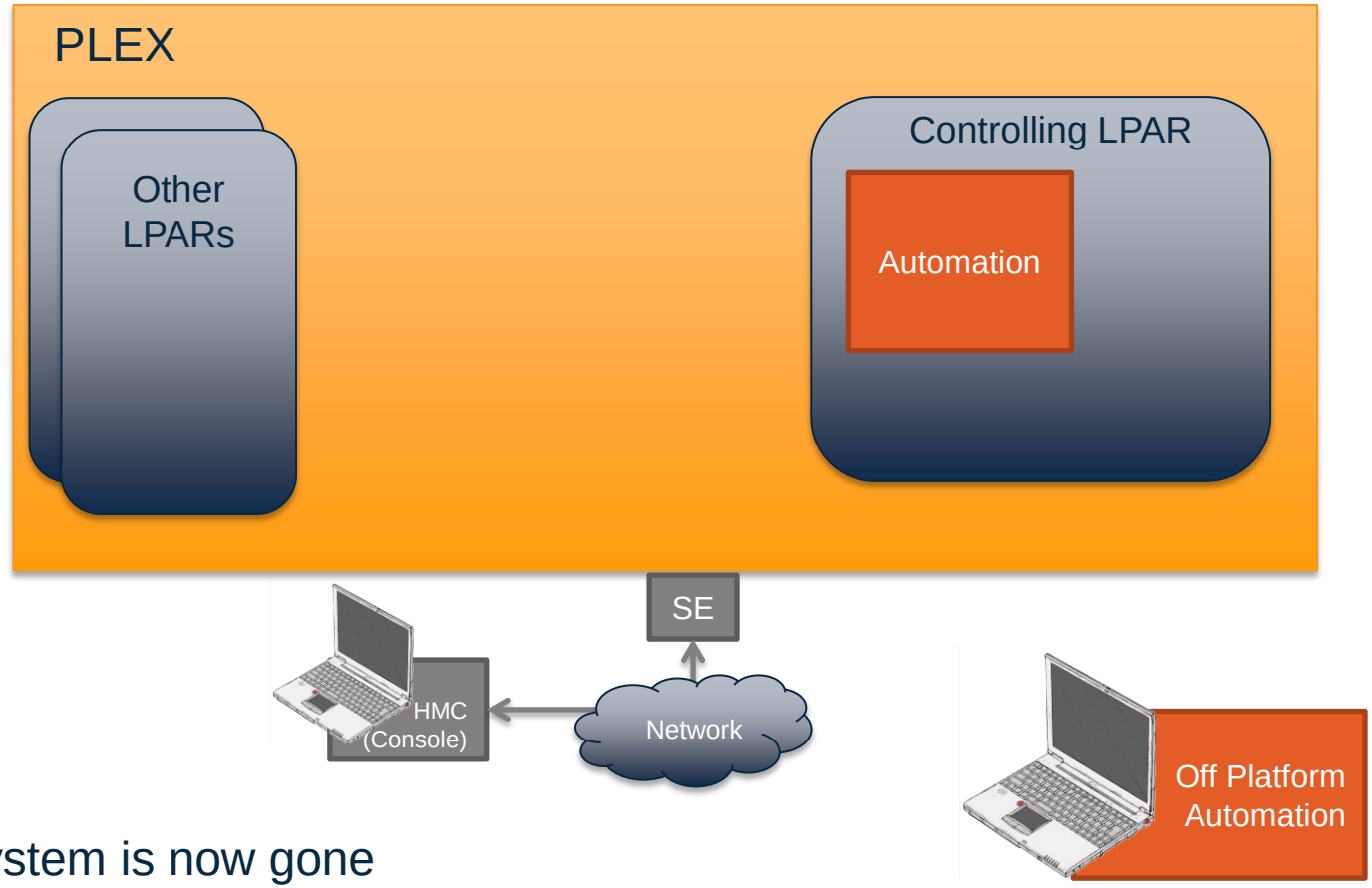
Initiate IPL -- Operator or automation
Have automation bring resources down cleanly

Use Case: Automated IPLs



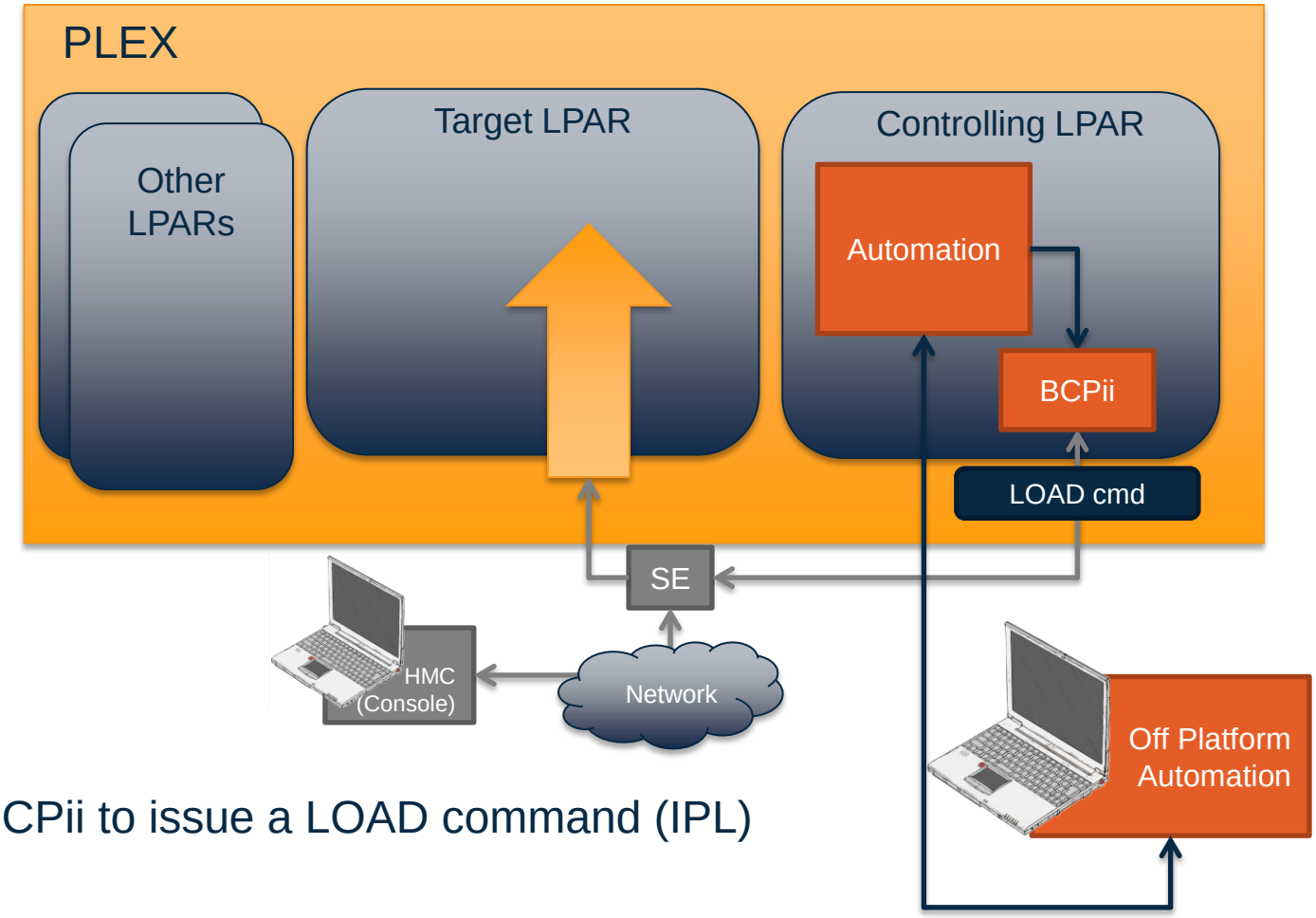
Use XCF to remove system from the sysplex
Use BCPii to issue reset with clear command

Use Case: Automated IPLs



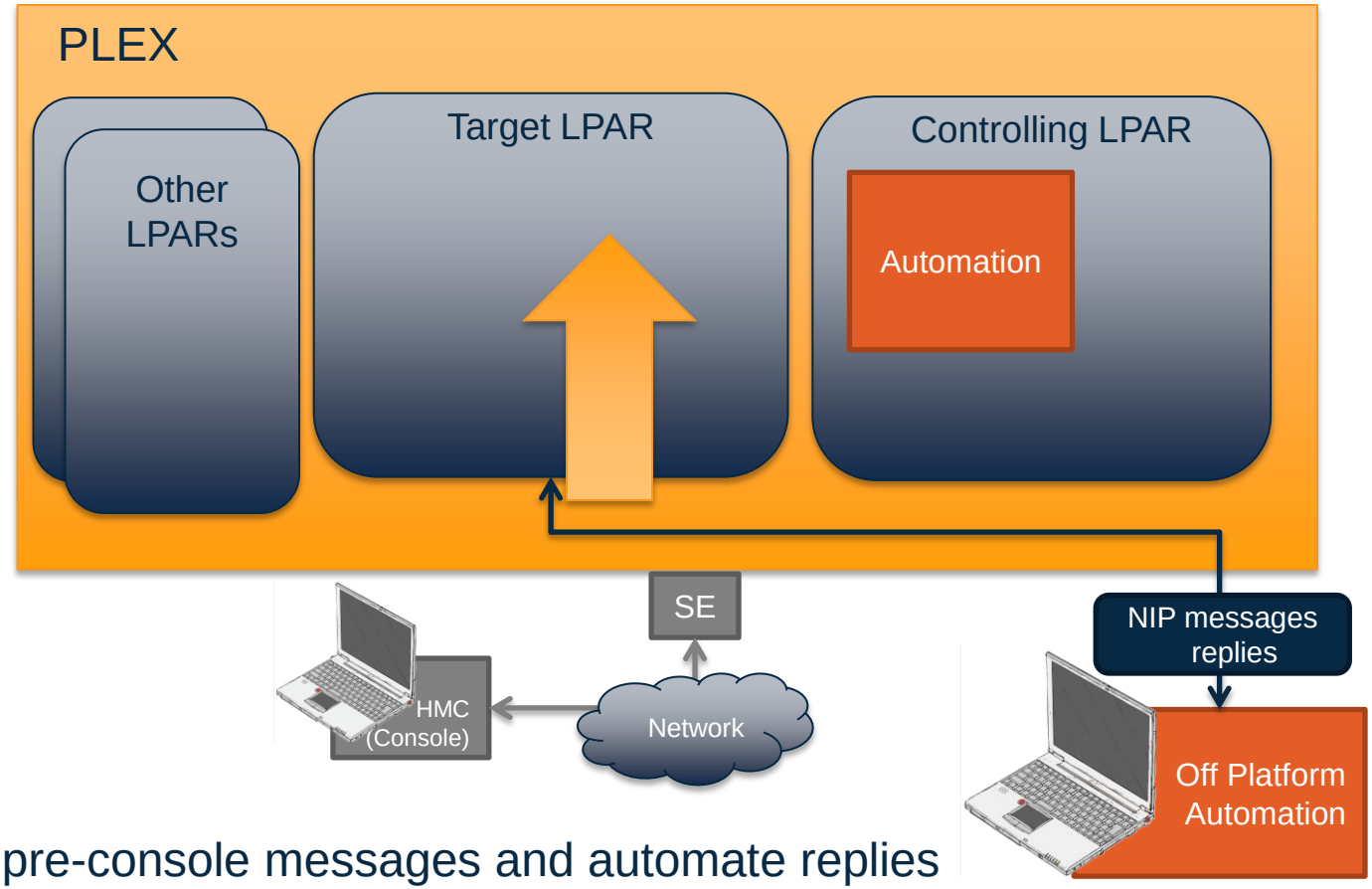
The system is now gone

Use Case: Automated IPLs



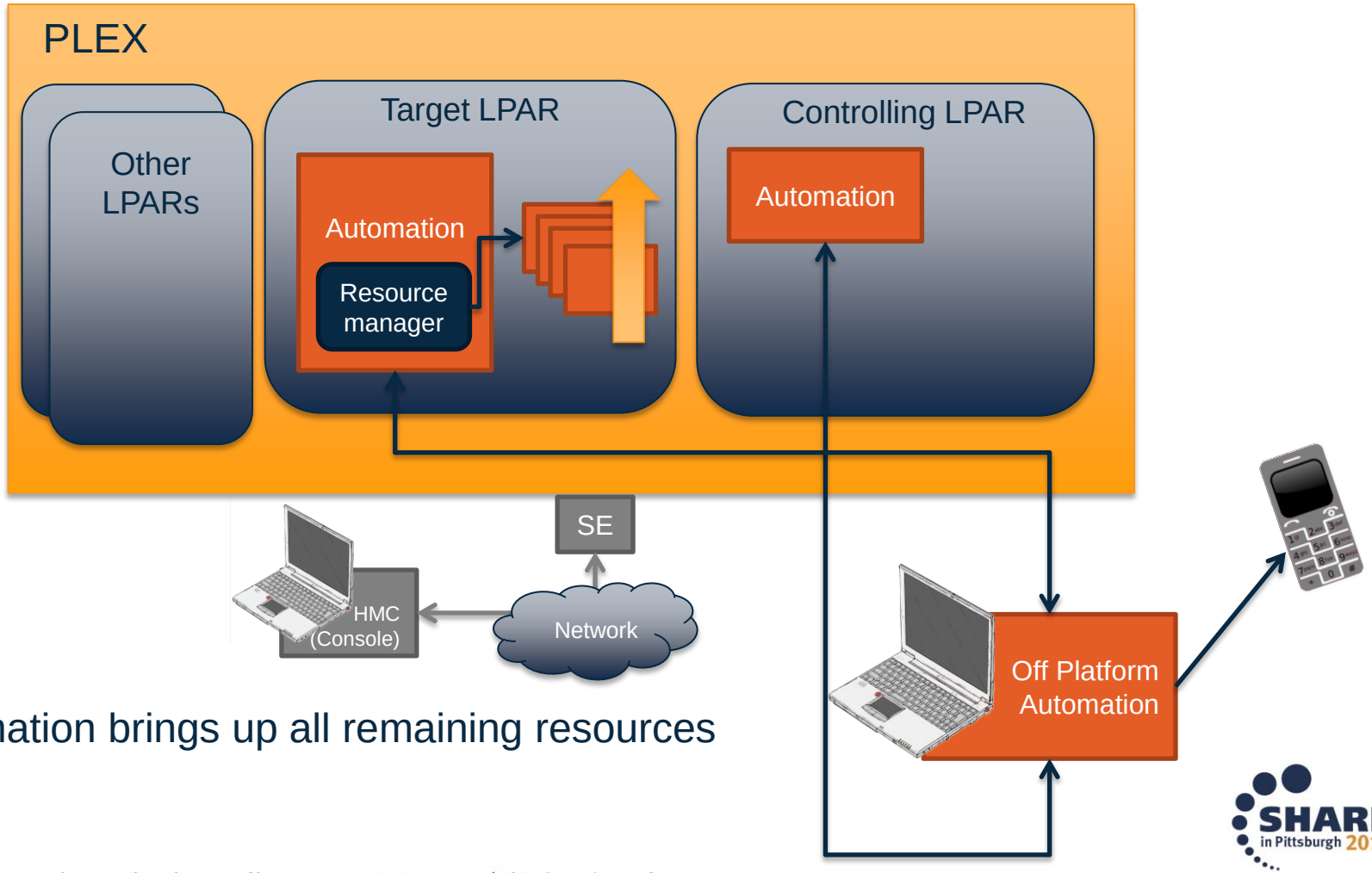
Use BCPii to issue a LOAD command (IPL)

Use Case: Automated IPLs



Catch pre-console messages and automate replies

Use Case: Automated IPLs



Automation brings up all remaining resources

Questions / Comments / Thoughts / Discussion

zachary.williams@ca.com

Legal Notice

THIS PRESENTATION IS FOR YOUR INFORMATIONAL PURPOSES ONLY. CA assumes no responsibility for the accuracy or completeness of the information. TO THE EXTENT PERMITTED BY APPLICABLE LAW, CA PROVIDES THIS DOCUMENT “AS IS” WITHOUT WARRANTY OF ANY KIND, INCLUDING, WITHOUT LIMITATION, ANY IMPLIED WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, OR NONINFRINGEMENT. In no event will CA be liable for any loss or damage, direct or indirect, in connection with this presentation, including, without limitation, lost profits, lost investment, business interruption, goodwill, or lost data, even if CA is expressly advised of the possibility of such damages.

Certain information in this presentation may outline CA’s general product direction. This presentation shall not serve to (i) affect the rights and/or obligations of CA or its licensees under any existing or future written license agreement or services agreement relating to any CA software product; or (ii) amend any product documentation or specifications for any CA software product. The development, release and timing of any features or functionality described in this presentation remain at CA’s sole discretion.

Notwithstanding anything in this presentation to the contrary, upon the general availability of any future CA product release referenced in this presentation, CA may make such release available (i) for sale to new licensees of such product; and (ii) in the form of a regularly scheduled major product release. Such releases may be made available to current licensees of such product who are current subscribers to CA maintenance and support on a when and if-available basis.

Additional Use Cases

- Change LPAR weights on a scheduled basis (time of day)
- Change LPAR weights based on performance thresholds.
- Turn on/off capacity on demand based on performance.
- Get hardware configuration information (attributes) and save it off platform for “big data” analysis.
- Changing power settings based on performance or DR system switch.
- Change activation profile to adjust how systems come back up (like a safe mode).
- Change weights/capacity to maintain SLAs (1 to 1, system to application).
- Change weights/capacity based on queue depth (1 to 1, system to application).