Init7

Network Configuration in the Real World
Standards, Surprises and 'Creative' Implementations

Jens Vogler

About Me

- 2020 – Started at Init7 as Software Engineer
- 2022 – Finished my Bachelor at ZHAW with my Thesis called “Network Infrastructure automation with NetBox”
- Trying to actually implement automation ever since ...
- Picked up woodworking as a hobby

OSS & BSS

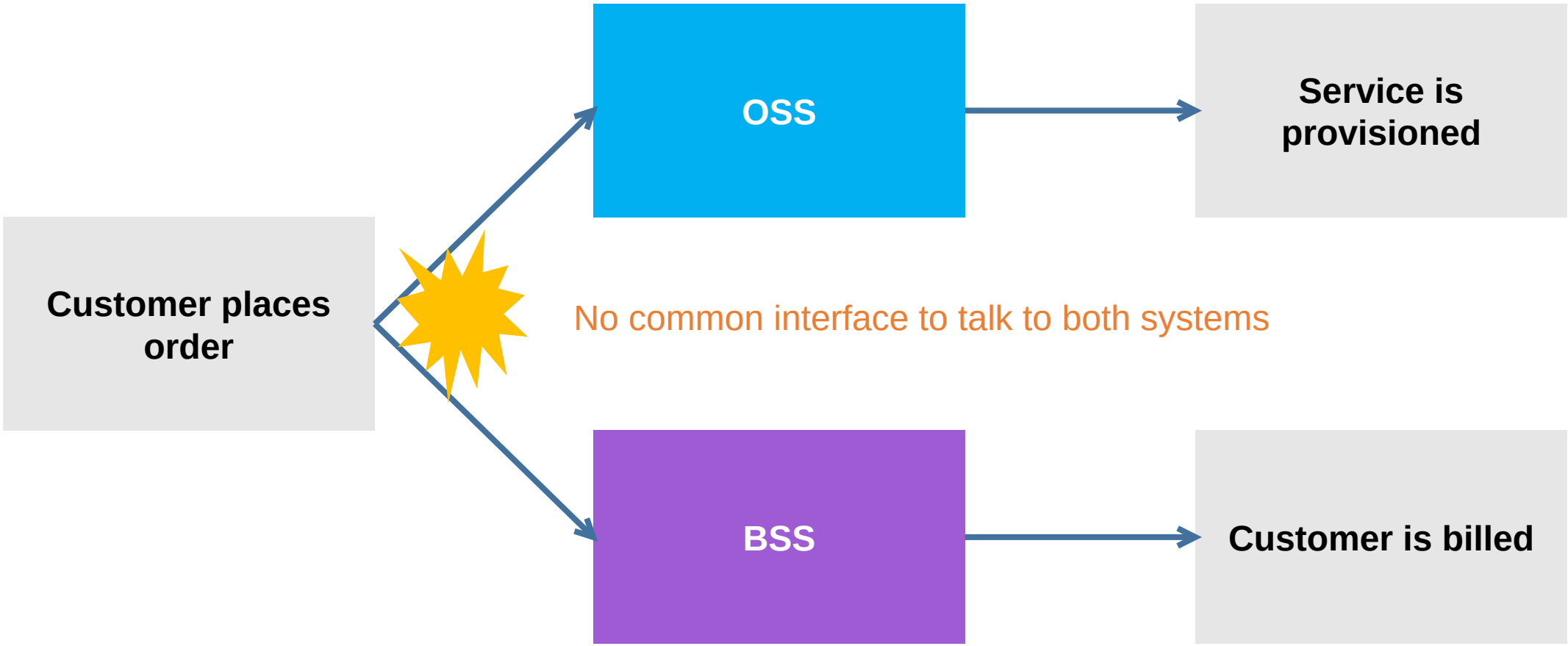
OSS - Operations Support System

- Network Inventory
- Network Operations

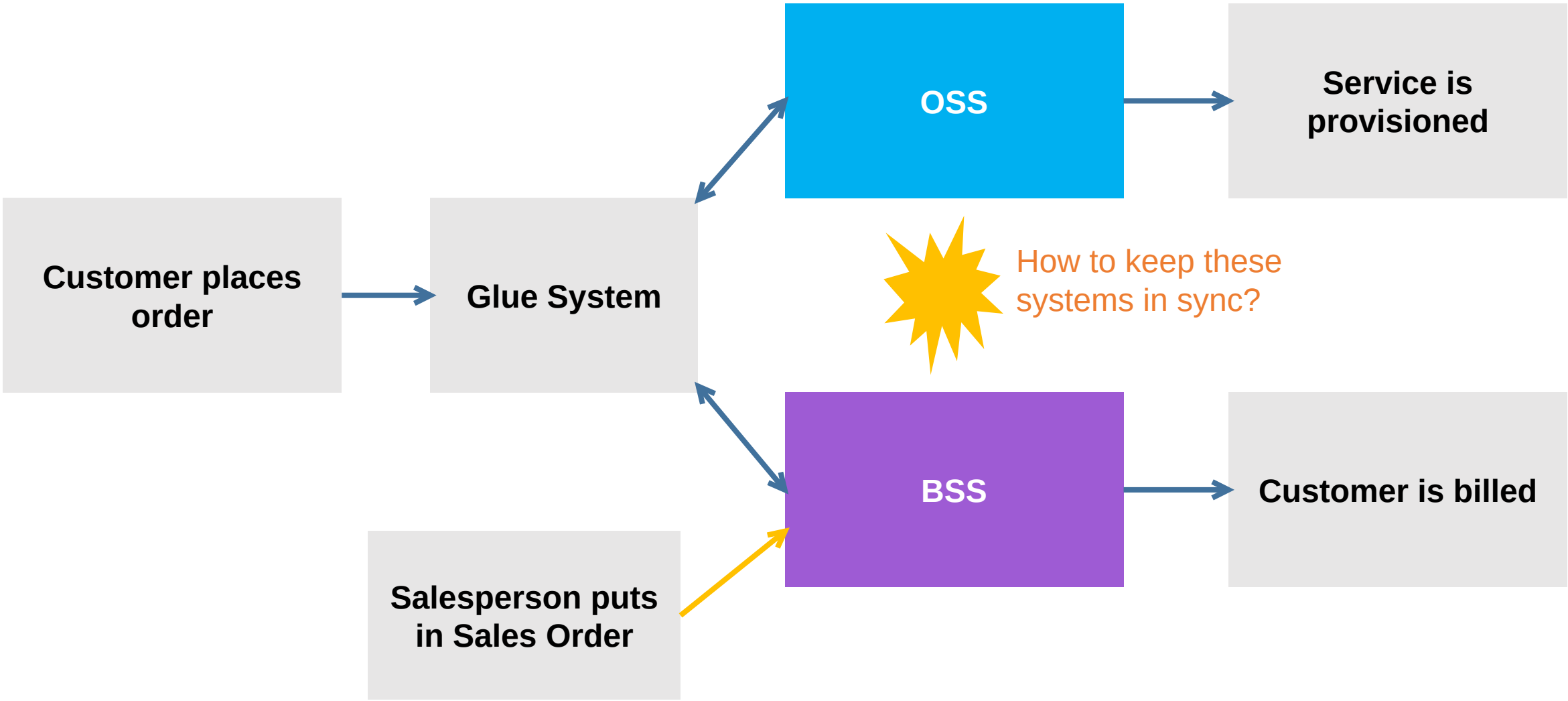
BSS - Business Support System

- CRM
- Billing

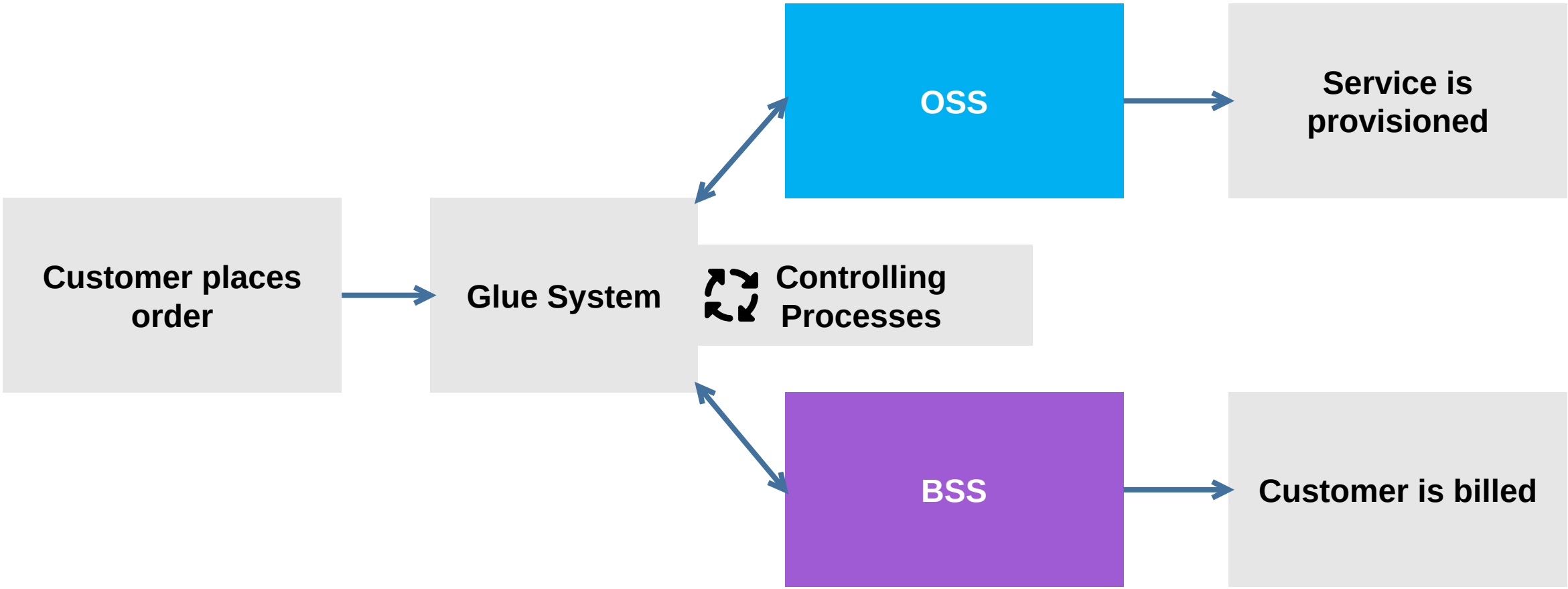
The Process


OSS
Service is provisioned
Customer places order
No common interface to talk to both systems
BSS
Customer is billed

The Process

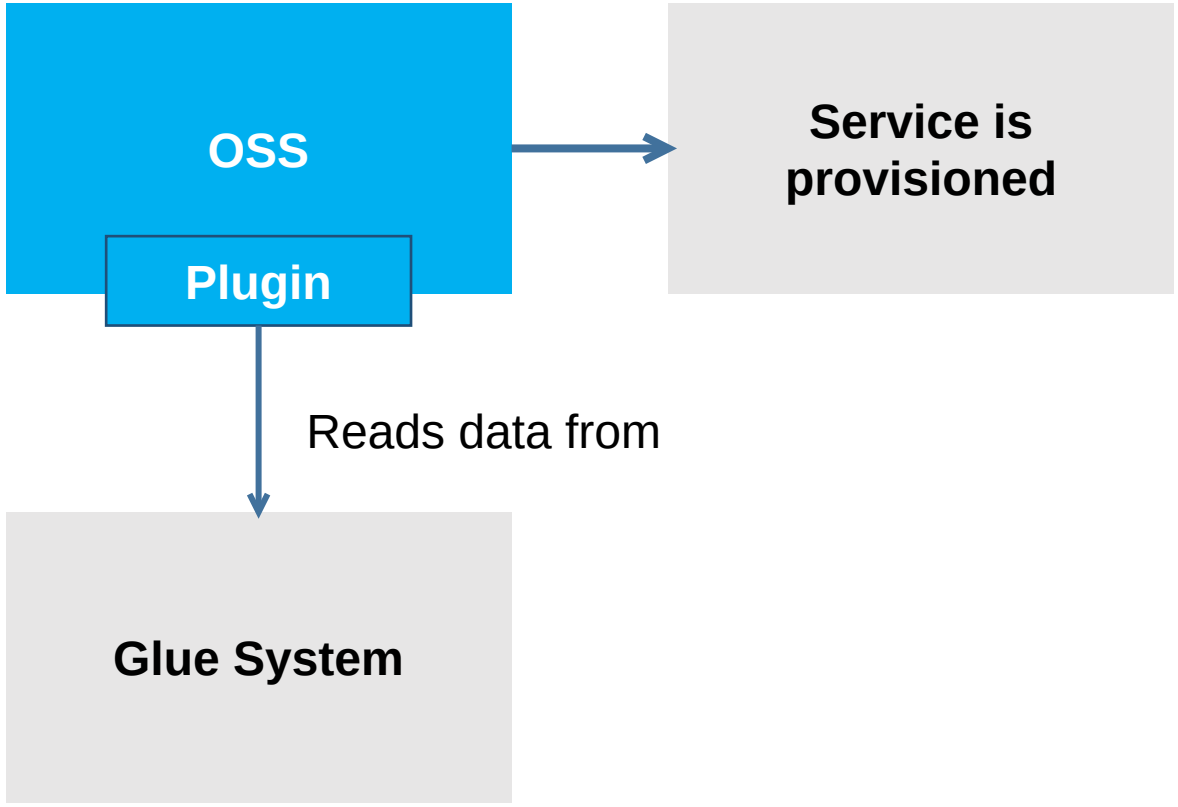

Customer places order
Glue System
OSS
Service is provisioned
How to keep these systems in sync?
BSS
Customer is billed
Salesperson puts in Sales Order

The Process


Customer places order
Glue System
Controlling Processes
OSS
BSS
Service is provisioned
Customer is billed

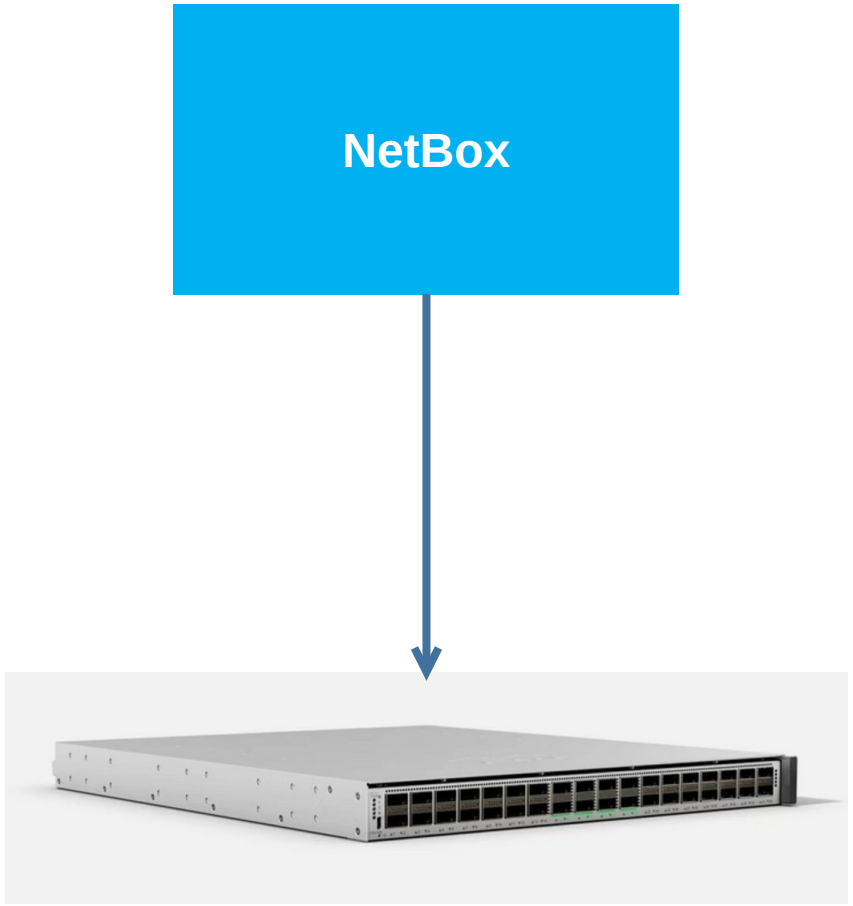
Service provisioning

- Do we have all the data we need?
  - Customer name on the interface description?



Getting the config onto the Device

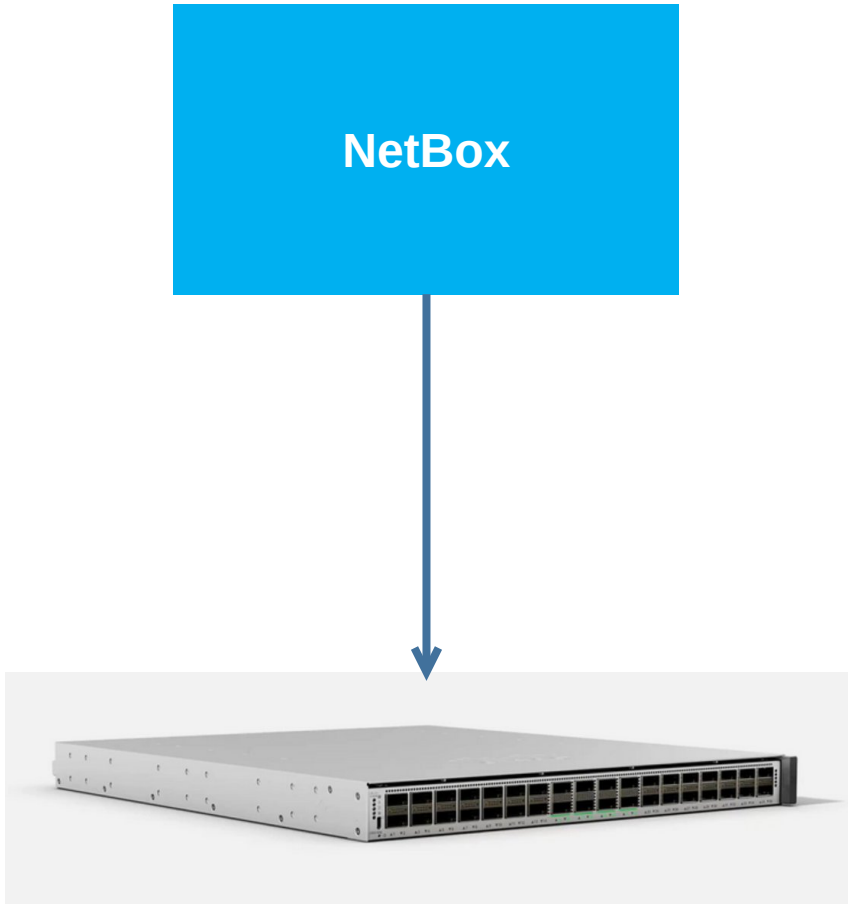
- NETCONF!
  - RFC 4741 from 2006
  - Open standard and vendor agnostic
  - Supports preparing, validating and atomic commit of configuration
  - Supports replacing specific sections of configuration
  - Supplemented by YANG to get a complete schema of possible configuration



Getting the config onto the Device

- So it’s easy right? Just write a template once for your datamodel
  - YANG Models defined by IETF are barely usable
  - OpenConfig tried to solve this, but vendor support is very limited

- Okay fine, so let’s write vendor specific templates instead...



So what does NETCONF “Support” really mean?

```
interface TwentyFiveGigE1/0/42
 description s1.480tst - 1/0/42
 switchport access vlan 700
 switchport mode access
 switchport protected
 switchport port-security maximum 5
 switchport port-security
 device-tracking attach-policy DT_UNTRUSTED
 ip access-group IGMP in
 ipv6 nd raguard
 ipv6 dhcp-ldra interface-id s1.480tst_1/0/42
 ipv6 dhcp guard
 storm-control broadcast level pps 1k
 storm-control multicast level pps 1k
 storm-control unknown-unicast level pps 1k
 storm-control action shutdown
 no lldp transmit
 no lldp receive
 spanning-tree bpdufilter enable
 ip igmp max-groups 7
 ip igmp filter 7
 ip dhcp snooping vlan 700 information option format-type
circuit-id string s1.480tst_1/0/42
```

So what does NETCONF “Support” really mean?

```
interface TwentyFiveGigE1/0/42
 description s1.480tst - 1/0/42
 switchport access vlan 700
 switchport mode access
 switchport protected
 switchport port-security maximum 5
 switchport port-security
 device-tracking attach-policy DT_UNTRUSTED
 ip access-group IGMP in
 ipv6 dhcp guard
 storm-control action shutdown
 no lldp transmit
 no lldp receive
 spanning-tree bpdufilter enable
 ip igmp max-groups 7
 ip igmp filter 7
 ip dhcp snooping vlan 700 information option format-type
circuit-id string s1.480tst_1/0/42
```

So what does NETCONF “Support” really mean?

```
@@ -1,23 +1,17 @@
 interface TwentyFiveGigE1/0/42
  description s1.480tst - 1/0/42
  switchport access vlan 700
  switchport mode access
  switchport protected
  switchport port-security maximum 5
  switchport port-security
  device-tracking attach-policy DT_UNTRUSTED
  ip access-group IGMP in
- ipv6 nd raguard
- ipv6 dhcp-ldra interface-id s1.480tst_1/0/42
  ipv6 dhcp guard
- storm-control broadcast level pps 1k
- storm-control multicast level pps 1k
- storm-control unknown-unicast level pps 1k
  storm-control action shutdown
  no lldp transmit
  no lldp receive
  spanning-tree bpdufilter enable
  ip igmp max-groups 7
  ip igmp filter 7
  ip dhcp snooping vlan 700 information option format-type
circuit-id string s1.480tst_1/0/42

```

Really?

```
<rpc-reply xmlns="urn:ietf:params:xml:ns:netconf:base:1.0" xmlns:nc="urn:ietf:params:xml:ns:netconf:base:1.0"
           message-id="urn:uuid:2abb4728-1f78-4f95-a724-b4aac2b11603">
 <rpc-error>
  <error-type>application</error-type>
  <error-tag>invalid-value</error-tag>
  <error-severity>error</error-severity>
  <error-message xml:lang="en">inconsistent value: Device refused one or more commands</error-message>
  <error-info>
   <severity xmlns="http://cisco.com/yang/cisco-ia">error_cli</severity>
   <detail xmlns="http://cisco.com/yang/cisco-ia">
    <bad-cli>
     <bad-command>no switchport mode access</bad-command>
     <error-location>15</error-location>
     <parser-response>Command rejected: Conflict with Port Security</parser-response>
     <parser-context>no description
      no switchport access vlan 700
      no switchport port-security maximum 5
      no switchport mode access
     </parser-context>
    </bad-cli>
   </detail>
  </error-info>
 </rpc-error>
</rpc-reply>
```

Back to pushing CLI commands...

- Well established tools - Ansible
- Network Engineers often already have some training

- Applying configuration is non-atomic
- Replacing sections reliably is impossible

```
@@ -1,23 +1,19 @@
 interface TwentyFiveGigE1/0/42
- description s1.480tst - 1/0/42
- switchport access vlan 700
- switchport mode access
- switchport protected
- switchport port-security maximum 5
+ description Easy7 / s1.480tst - 1/0/42
+ switchport private-vlan host-association 701 101
+ switchport mode private-vlan host
+ switchport port-security maximum 1
  switchport port-security
+ switchport port-security mac-address sticky
+ switchport port-security violation restrict
  device-tracking attach-policy DT_UNTRUSTED
- ip access-group IGMP in
+ ip arp inspection limit rate 25
  ipv6 nd raguard
- ipv6 dhcp-ldra interface-id s1.480tst_1/0/42
- ipv6 dhcp guard
  storm-control broadcast level pps 1k
  storm-control multicast level pps 1k
  storm-control unknown-unicast level pps 1k
  storm-control action shutdown
  no lldp transmit
  no lldp receive
  spanning-tree bpdufilter enable
- ip igmp max-groups 7
- ip igmp filter 7
- ip dhcp snooping vlan 700 information option format-type circuit-id string s1.480tst_1/0/42
+ ip dhcp snooping vlan 701 information option format-type circuit-id string s1.480tst_1/0/42
```

Learnings & Takeaways

- Current generation software seems to catch up with NETCONF support – though verifying and testing is still necessary
  - Model based configuration system is key
- Tooling around NETCONF, RESTCONF & YANG has room for improvement
  - Cisco YANG Suite – Powerful but slow, glitchy and closed-source
  - libnetconf2, libyang – Low level
  - Pyang, ncclient – Probably the easiest for automation
- Single source of truth doesn’t solve all problems
  - Escape hatches are often necessary to fix urgent issues
  - Attempting to represent reality through abstract data models will always contain gaps
  - Once synchronization is introduced, divergence is possible
- You’ll always encounter legacy systems where workarounds are needed

Thank you!

Init7