

# NML Topology in NSI

Freek Dijkstra and Jeroen van der Ham

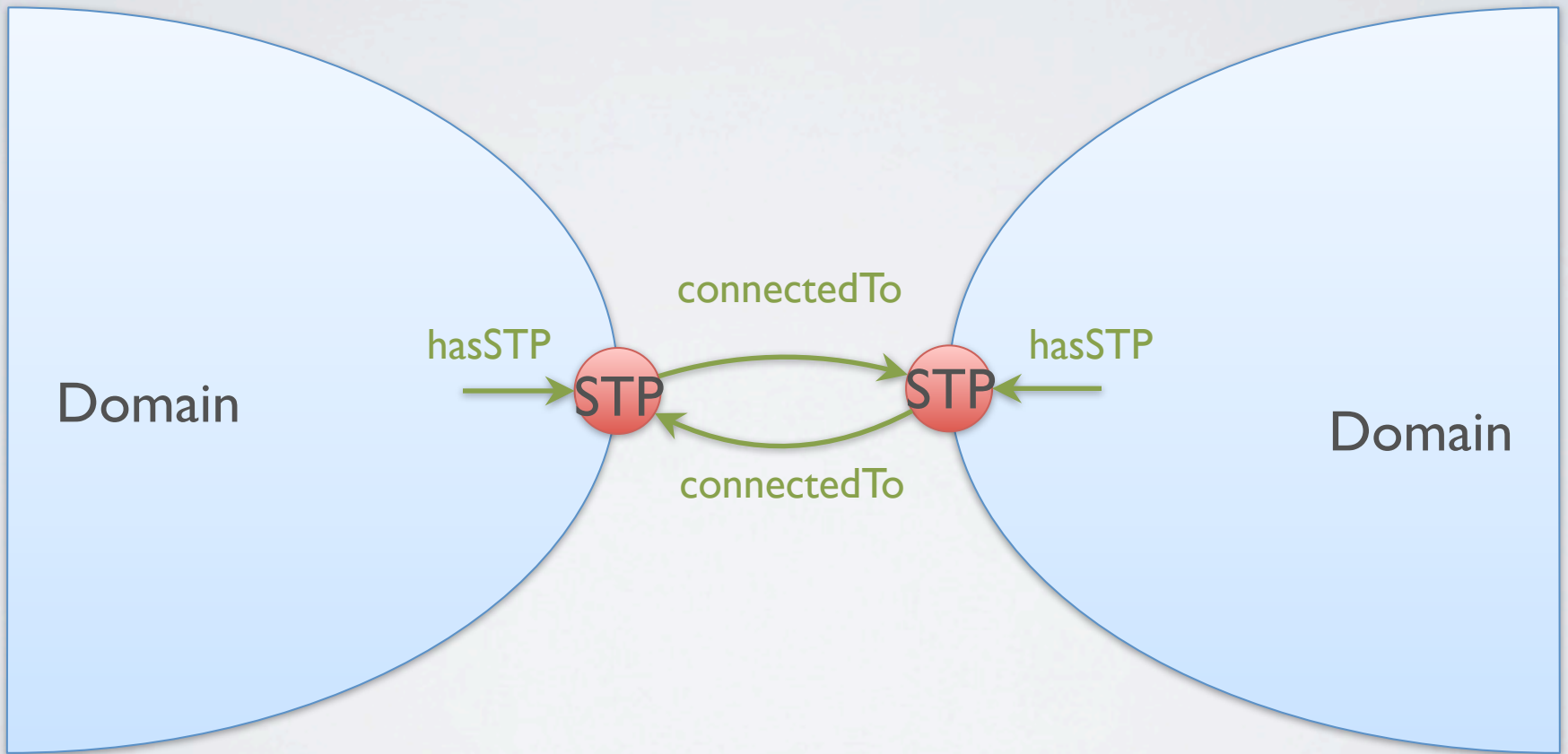
# NSI

- STP Service Termination Point.
  - connectedTo relation to form an SDP with another STP
- NSNetwork Network Service Network
  - hasSTP to define STP containment
  - locatedAt to define a location of a network
- Location
  - lat, long define GPS coordinates
- NSA Network Service Agent
  - managedBy to relate NSNetwork to NSA
  - adminContact to describe contacts for the administrator
  - csProviderEndpoint to define the URL at which the NSA is reachable
  - connectedTo to related two NSA

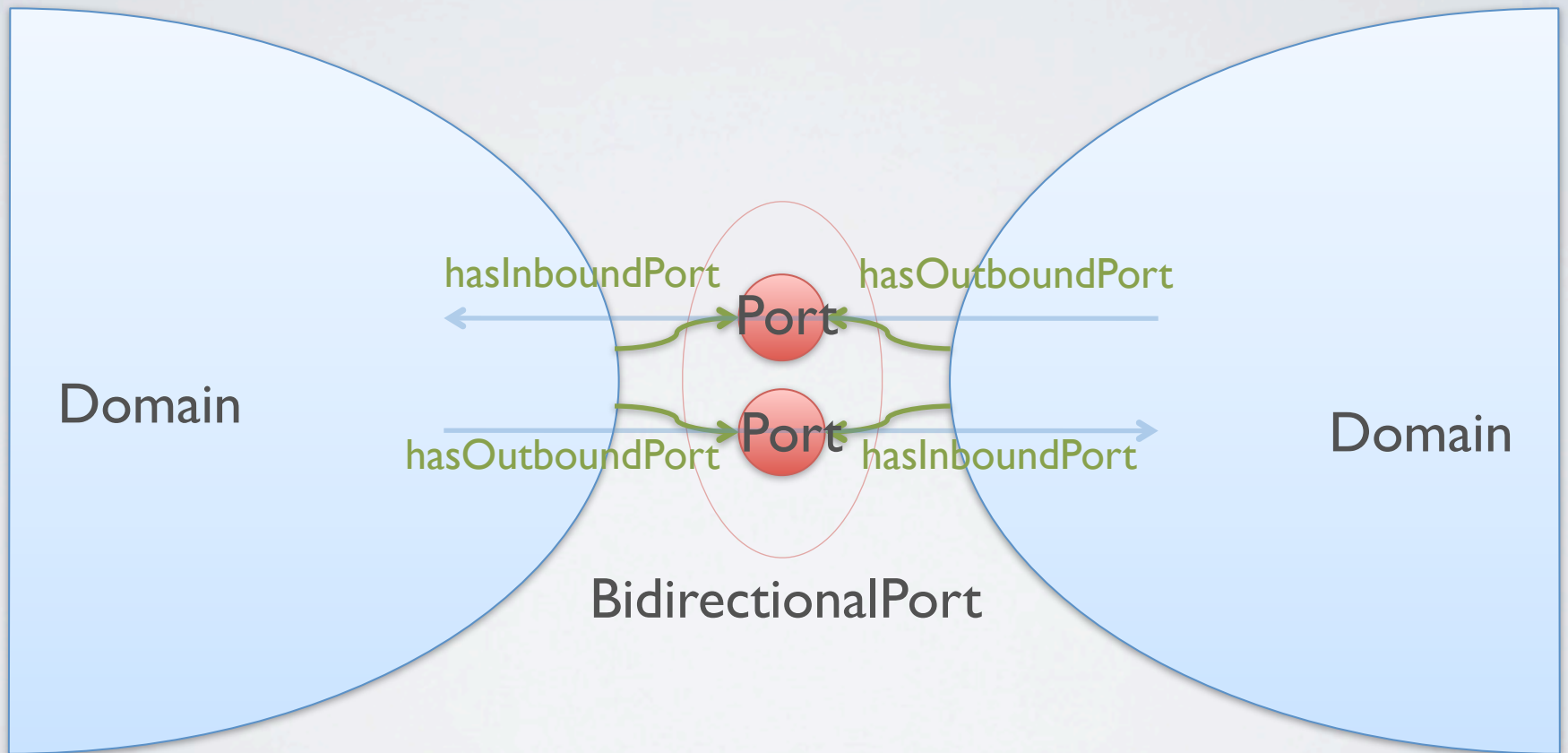
# Move Topology from NSI to NML

NSI	NML
<b>2x nsi:STP</b>	<b>2x nml:Port + nml:BidirectionalPort</b>
nsi:connectedTo	nml:alias (optional)
<b>nsi:NSNetwork</b>	<b>nml:Topology</b>
nsi:hasSTP	nml:hasPort
nsi:locatedAt	nml:locatedAt
<b>nsi:Location</b>	<b>nml:Location</b>
nsi:lat, nsi:long	nml:lat, nml:long, nml:unlocode
<b>nsi:NSA</b>	<b>nsi:NSA</b>
nsi:managedBy	nsi:managedBy
nsi:adminContact	nsi:adminContact
nsi:csProviderEndpoint	nsi:csProviderEndpoint
nsi:connectedTo	nsi:connectedTo

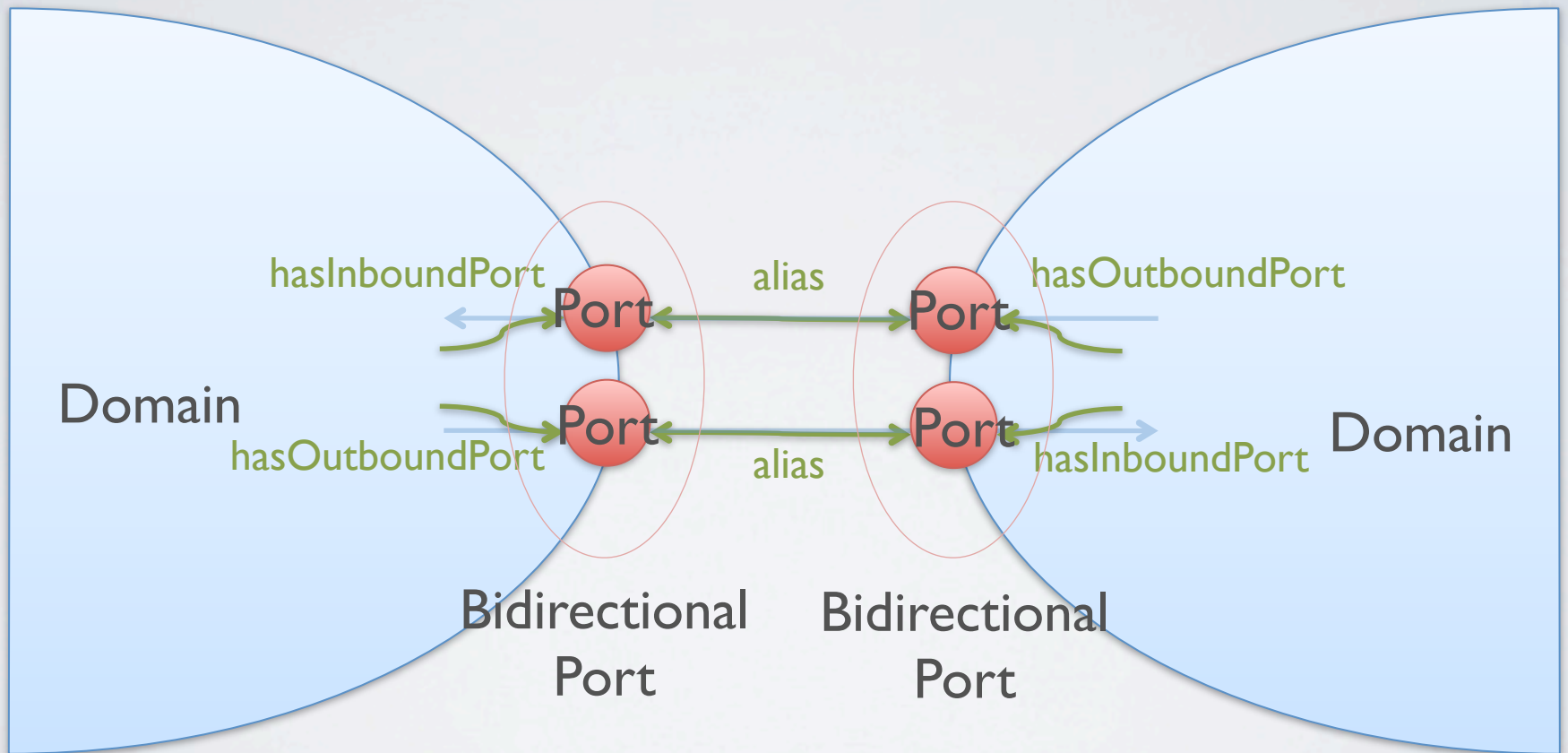
# NSI – SDP



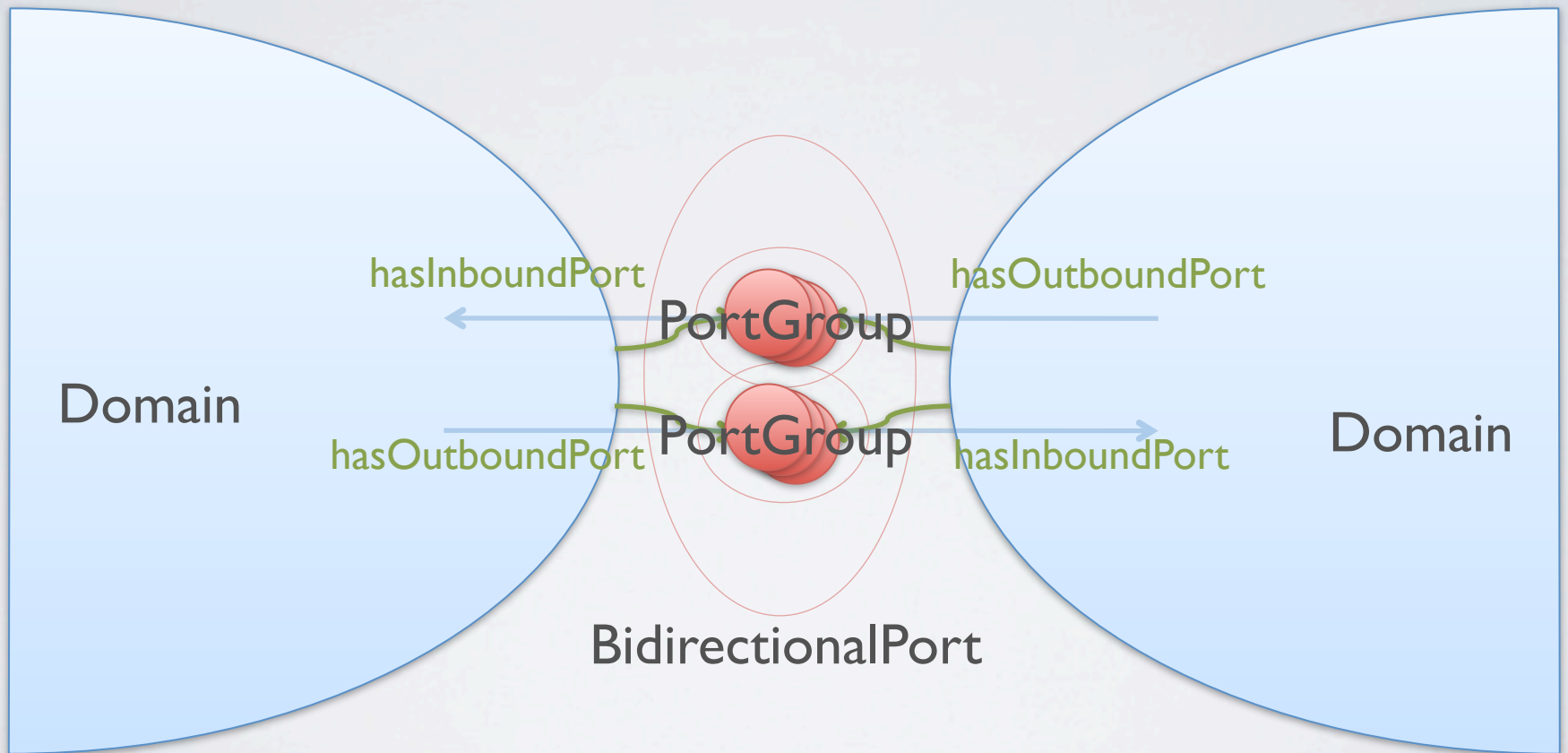
# NML – Port



# NML – Port with Aliases



# NML – PortGroup



# Identifiers

- Globally Unique
- Persistent (don't change, don't get re-assigned)



# Resource Names

urn:ogf:network:<DNSname>:<YYYYdate+>:<opaque>

# Resource Names

urn:ogf:network:<DNSname>:<YYYYdate+>:<opaque>



Unique prefix of URN-assigning organisation

# Identifying Ports and Labels

- Networks connections can be specified by a label (e.g.VLAN)
- NSI provisions network connections
- NSI request must be able to contain label-type and value
- NSI request must be able to contain a possible set of labels
- Network topologies should have an efficient representation
- Monitoring (et al.) needs a unique identifier per Port

# Port Groups Proposals

```
<nml:PortGroup id="urn:ogf:network:netherlight.net:2010:Asd001a-ome24:1-5-4:trunk:out">
  <nml:label>
    <nml:parameter name="type">vlan</nml:parameter>
    <nml:parameter name="set">100-110,210</nml:parameter>
  </nml:label>
</nml:PortGroup>
```

```
<nml:PortGroup id="urn:ogf:network:netherlight.net:2010:Asd001a-ome24:1-5-4:trunk:out">
  <nml:label>
    <nml:parameter name="type">vlan</nml:parameter>
    <nml:parameter name="values">
      <nml:parameter name="range">
        <nml:parameter name="start">0</nml:parameter>
        <nml:parameter name="end">4000</nml:parameter>
      </nml:parameter>
    </nml:parameter>
  </nml:label>
</nml:PortGroup>
```

# Port Group with Explicit Ports

```
<nml:PortGroup id="urn:ogf:network:netherlight.net:2010:Asd001a-ome24:1-5-4:trunk:out">
  <nml:Port idRef="urn:ogf:network:netherlight.net:2010:Asd001a-ome24:1-5-4:vlan11:out">
    <nml:label>
      <nml:parameter name="type">vlan</nml:parameter>
      <nml:parameter name="value">11</nml:parameter>
    </nml:label>
  </nml:Port>
  <nml:Port idRef="urn:ogf:network:netherlight.net:2010:Asd001a-ome24:1-5-4:vlan15:out"/>
    <nml:label>
      <nml:parameter name="type">vlan</nml:parameter>
      <nml:parameter name="value">15</nml:parameter>
    </nml:label>
  </nml:Port>
  <nml:Port idRef="urn:ogf:network:netherlight.net:2010:Asd001a-ome24:1-5-4:vlan42:out"/>
    <nml:label>
      <nml:parameter name="type">vlan</nml:parameter>
      <nml:parameter name="value">42</nml:parameter>
    </nml:label>
  </nml:Port>
</nml:PortGroup>
```

# Identifying a PortGroup

urn:ogf:network:netherlight.net:2010:Asd001a-  
ome24:1-5-4:trunk:out

# Identifying a Port by PortGroup + Label

```
urn:ogf:network:netherlight.net:2010:Asd001a-  
ome24:1-5-4:trunk:out?vlan=1781
```

# Identifying a Port with it's own URN

```
urn:ogf:network:netherlight.net:2010:Asd001a-  
ome24:1-5-4:trunk:out:vlan1781
```



# Proposal for NSI – NML

- Use NML Topology and PortGroups to describe Topology instead of NSNetworks and STPs
- Freek and Jeroen will translate the current topology into NML