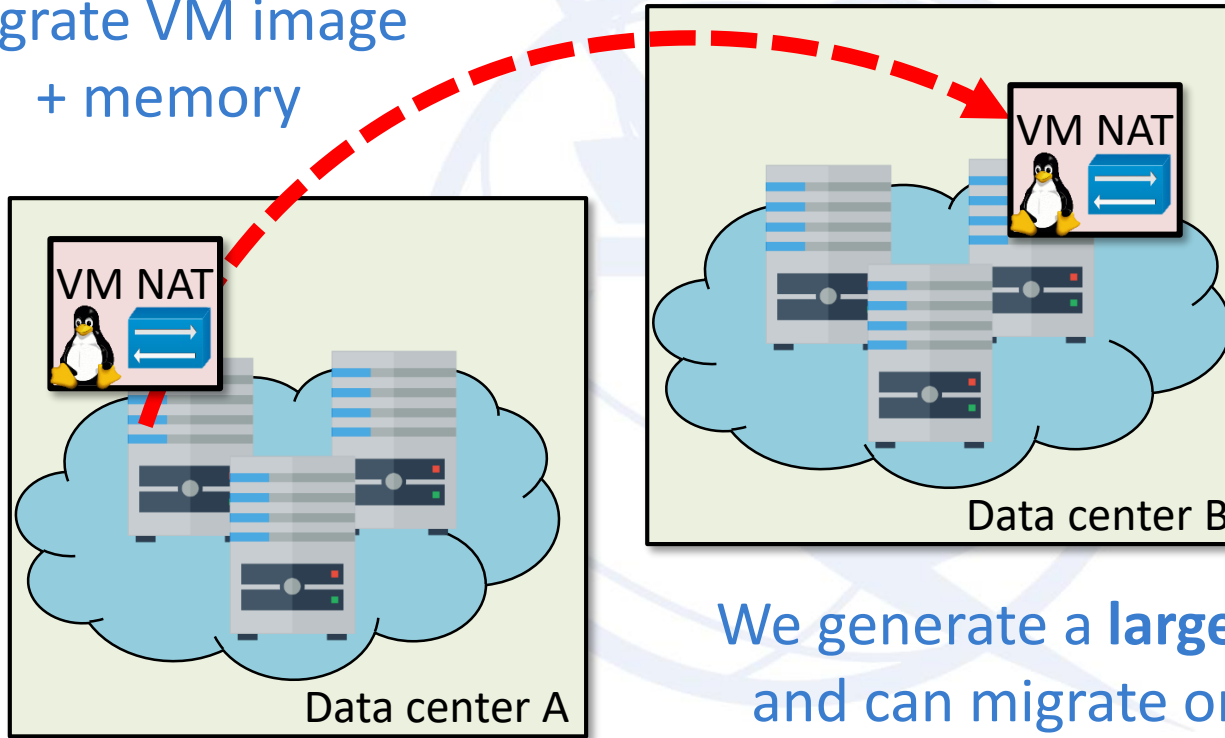# Live Migration of 5G Services between Heterogeneous Infrastructure Domains

L. Pesando, A. Manzalini (TIM), F. Risso, I. Cerrato (Politecnico di Torino)
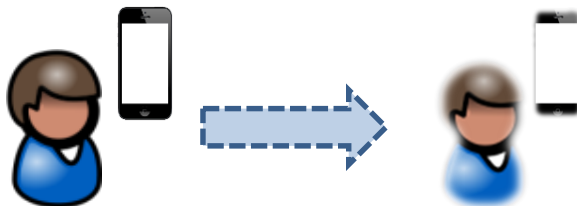
For more mobile, agile and resilient services

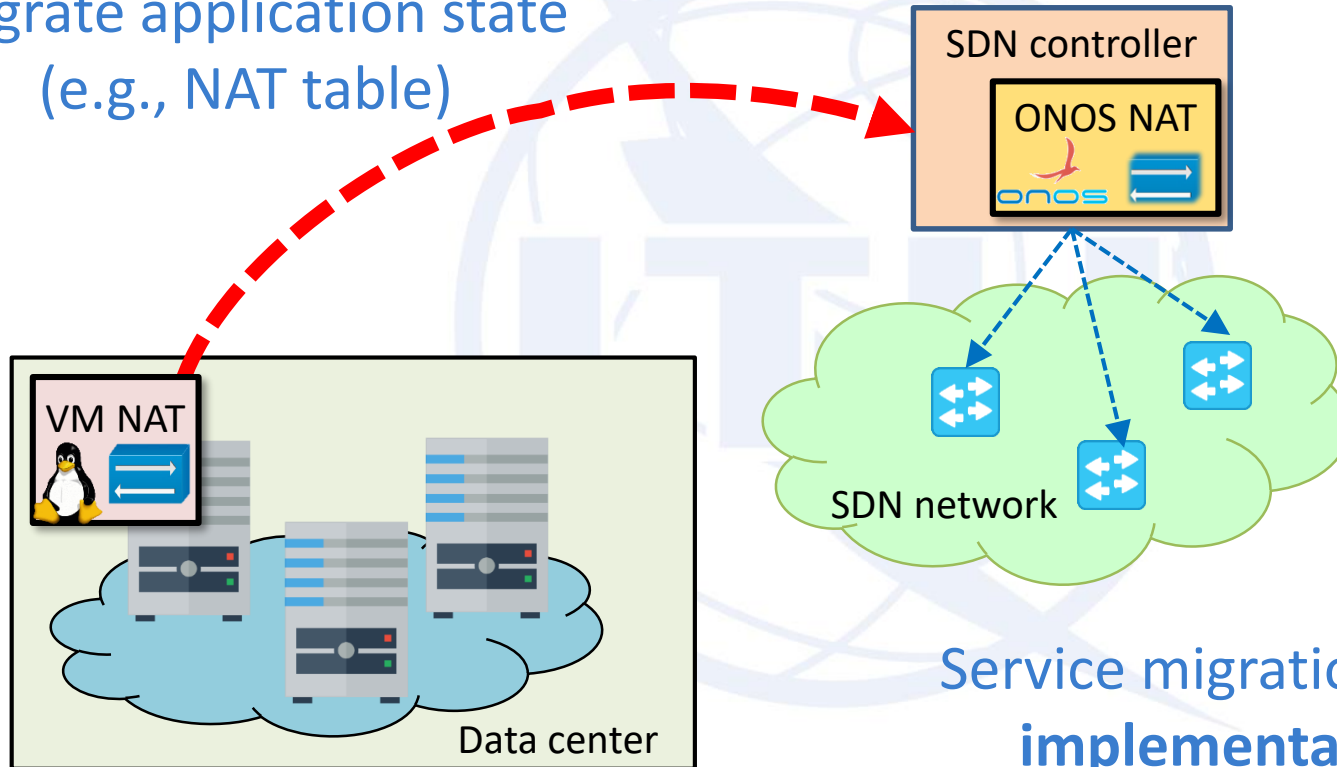# Current live migration technology

Migrate VM image
+ memory

VM NAT

Data center A

VM NAT

Data center B

We generate a **large amount of traffic**
and can migrate only data between
identical service instances

Migrate *state*, not *everything*!

Migrate application state
(e.g., NAT table)

SDN controller

ONOS NAT

VM NAT

Data center

SDN network

Service migration now becomes
implementation-agnostic
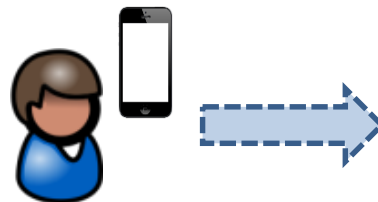
The key? A common Data Model

The essential information of each service is captured by a **data model**, which is used to transfer the actual **state** of the app
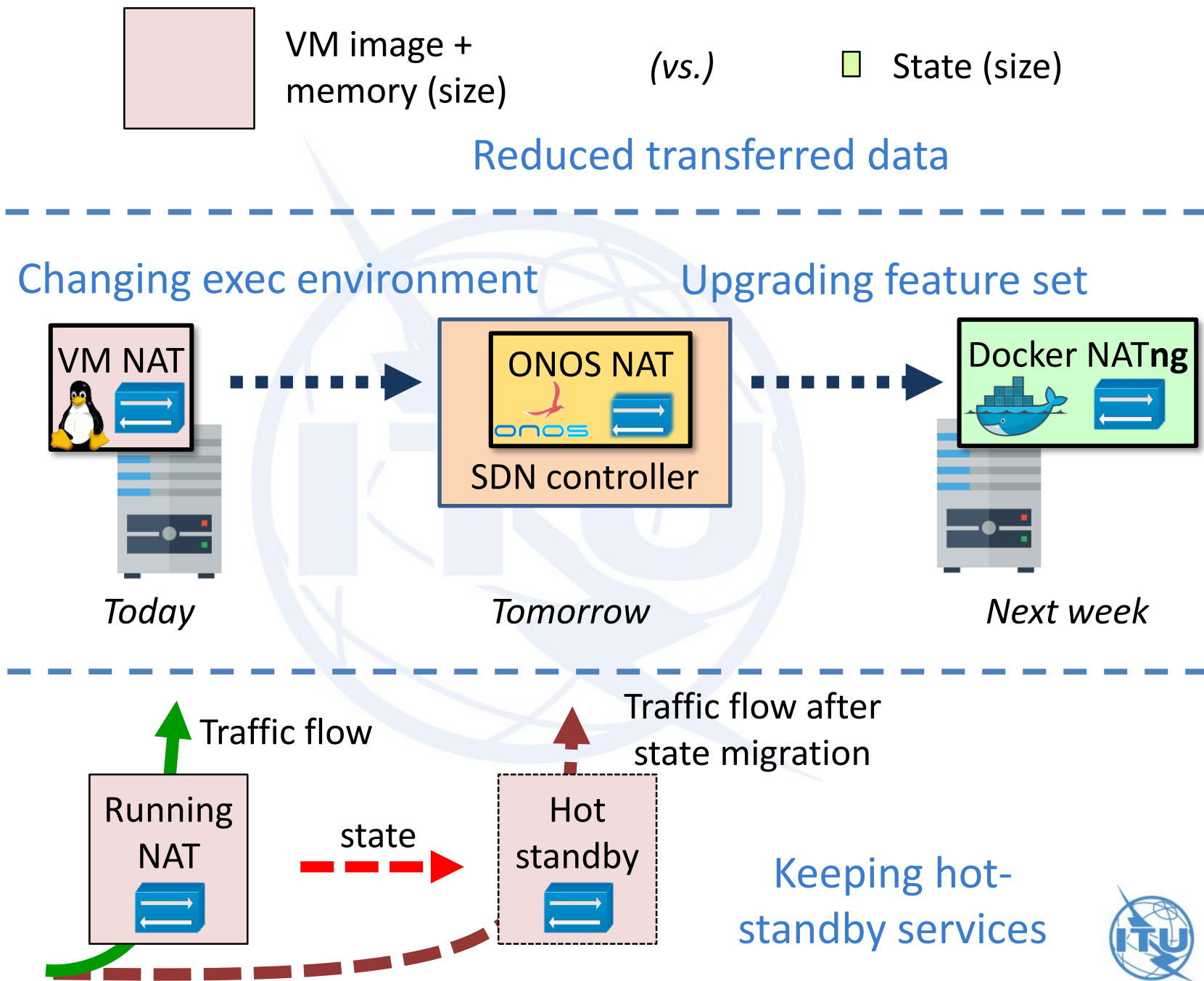
Implementation-agnostic

NAT

```
…
list nat-session {
  leaf protocol {
    type proto-type
  }
  leaf src_address {
    description "Source Address";
    type inet:ip-address;
  }
  leaf src_port {
    description "Source Port";
    type port-type;
  }
  leaf translated_address {
    description "Translated address";
    type inet:ip-address;
  }
  leaf translated_port {
    description "Destination port";
    type port-type;
  }
}
```

```
{
  "nat-session" : [
  {
    "protocol": "TCP",
    "src_address": "10.0.0.1",
    "src_port": "2526",
    "translated_address":
"130.192.225.79",
    "translated_port": "5678"
  },
  …
  ]
}
```

VM NAT

ONOS NAT

onos

Benefits

VM image + memory (size) *(vs.)* ☐ State (size)

Reduced transferred data

Changing exec environment   Upgrading feature set

VM NAT → ONOS NAT / SDN controller → Docker NAT**ng**

Today   *Tomorrow*   *Next week*

Traffic flow

Traffic flow after state migration

Running NAT — state → Hot standby

Keeping hot-standby services

Next Steps

- Generalize the approach with additional network functions

- Investigate possible issues when a portion of the state is shared with the operating system