Scenario

Software architecture

The software architecture is composed by three pieces:

- **Ancillary SIB**: stores informations about Virtual SIBs and remote SIBs to make them reachable;
- **Virtual SIB**: it has a public IP address and so it is a reachable SIB (not a real one, but we'll talk about it later);
- **Publisher**: it is a software agent that resides on the real SIB's host and interacts with the virtual SIB.

Virtual SIB

It is a black box. A KP can interact with the virtual SIB without noticing that it is not a one. It's just a process that acts as a SIB but...

The main purpose of the Virtual SIB it's just routing requests from the KPs to the real SIBs and confirms from the real SIBs to the KPs. Not only. The Virtual SIB examinates messages to mantain coherence and remove duplicates.

Publisher

The real sib, to communicate with the virtual SIB, needs a software agent that does the trick. This software component is the **publisher** that act as KP to the real SIB and communicates directly with the Virtual SIB.

Publisher

How does he inform the Virtual SIB about its presence? With a SSAP message that we defined:

```
<SSAP_message>
<node_id>a49d259b-e26d-40b4-a169-af1befe367cf-03cc02f9-dc7d-4462-974d-0a9d88b86416</node_id>
<space_id>X</space_id>
<transaction_type>REGISTER</transaction_type>
<message_type>REQUEST</message_type>
<transaction_id>?</transaction_id>
</SSAP_message>
```

The virtual SIB replies with the following message:

```
<SSAP_message>
<node_id>a49d259b-e26d-40b4-a169-af1befe367cf-03cc02f9-dc7d-4462-974d-0a9d88b86416</node_id>
<space_id>X</space_id>
<transaction_type>REGISTER</transaction_type>
<message_type>CONFIRM</message_type>
<transaction_id>?</transaction_id>
</SSAP_message>
```

Join / Leave

Insert

Remove

Query

Subscription

Unsubscribe

Ancillary SIB