



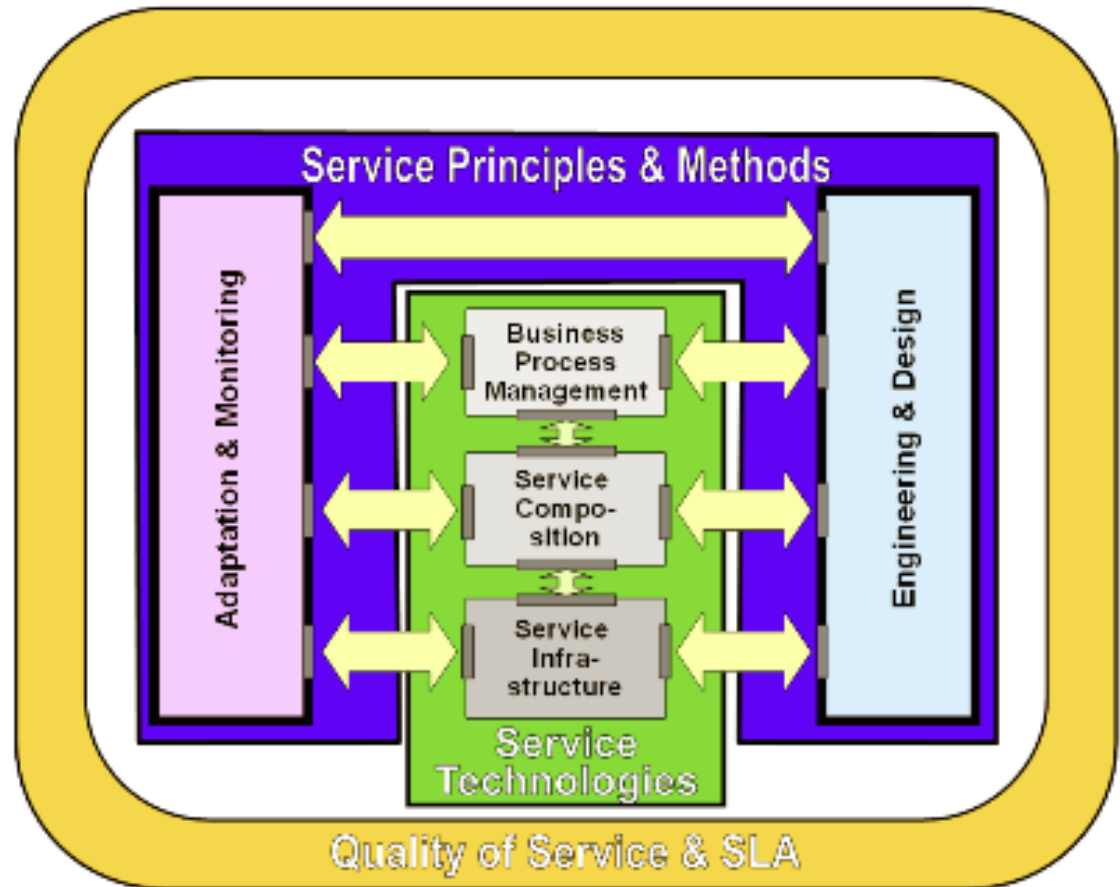
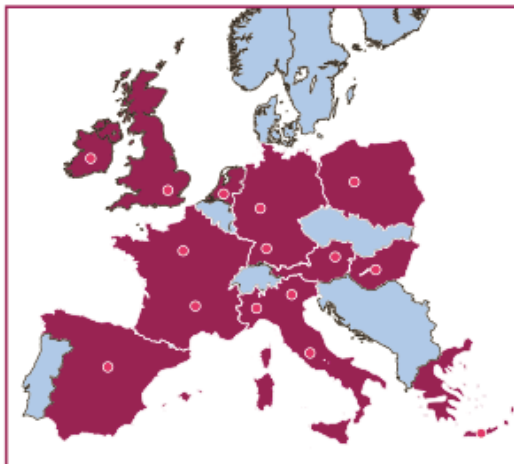
Autonomic Resource Virtualization in Cloud-like Environments

A. Kertész, G. Kecskeméti, I. Brandic
MTA SZTAKI, TUW

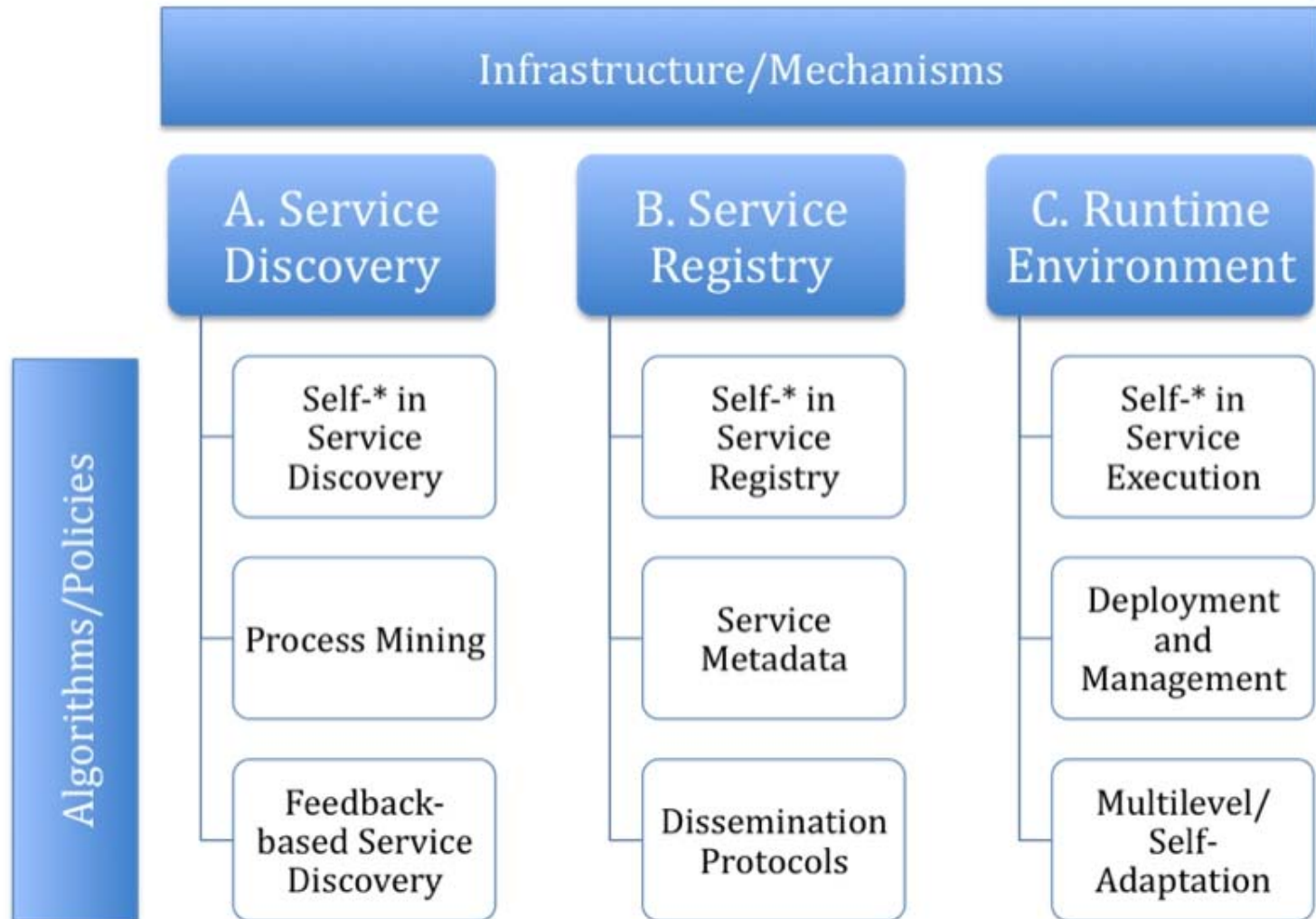
Joint EGEE and EDGeS Summer School on Grid Application Support
July 4, 2009, Budapest, Hungary

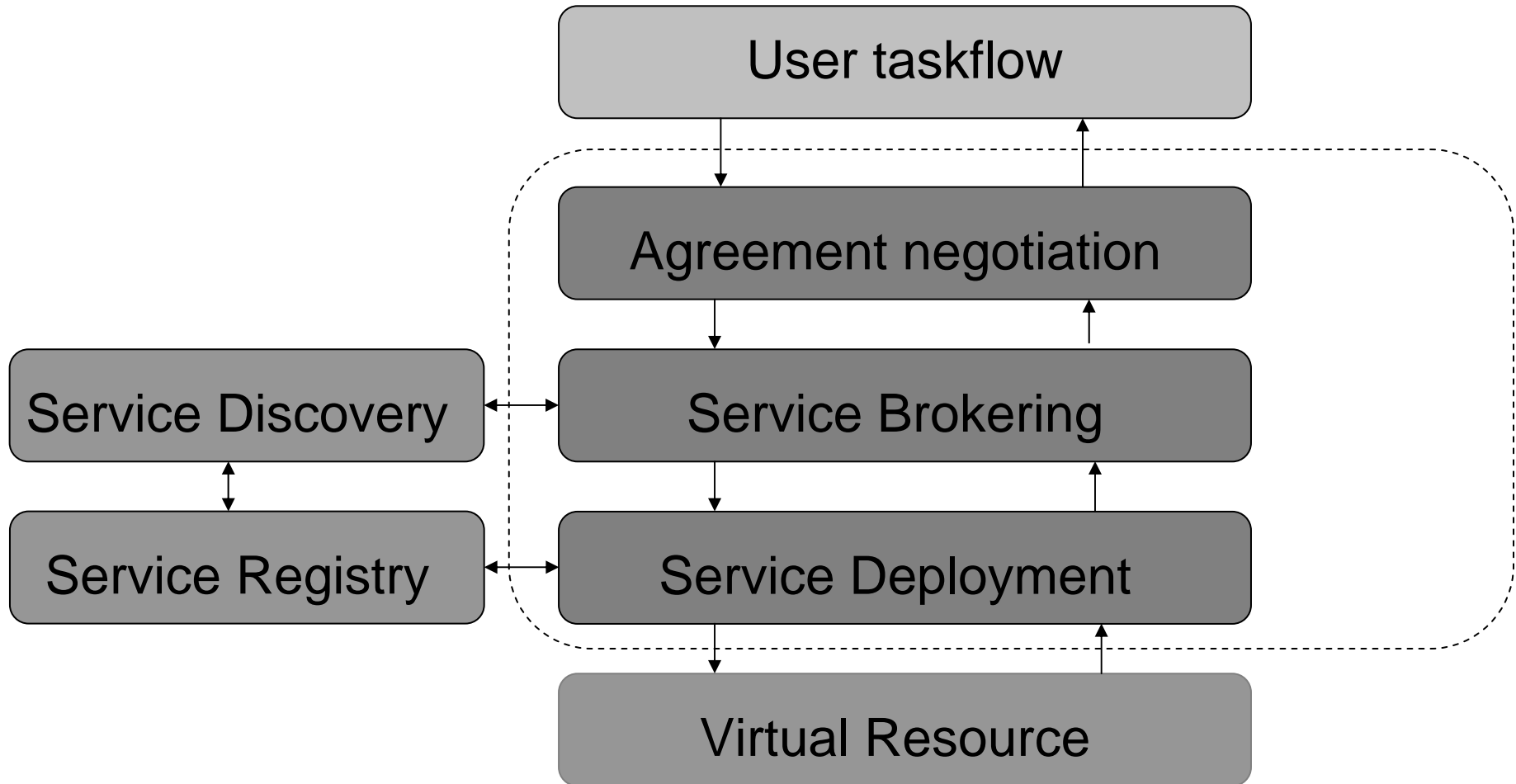
- What is S-Cube? Where are we in S-Cube?
- **SRV**: an architecture for SLA-based resource virtualization that provides an extensive solution for executing user applications in **Cloud-like** infrastructures
- **Autonomic** components of the architecture for enabling self-management in agreement negotiation, service brokering and deployment using virtualization

The Software Services and Systems Network = S³

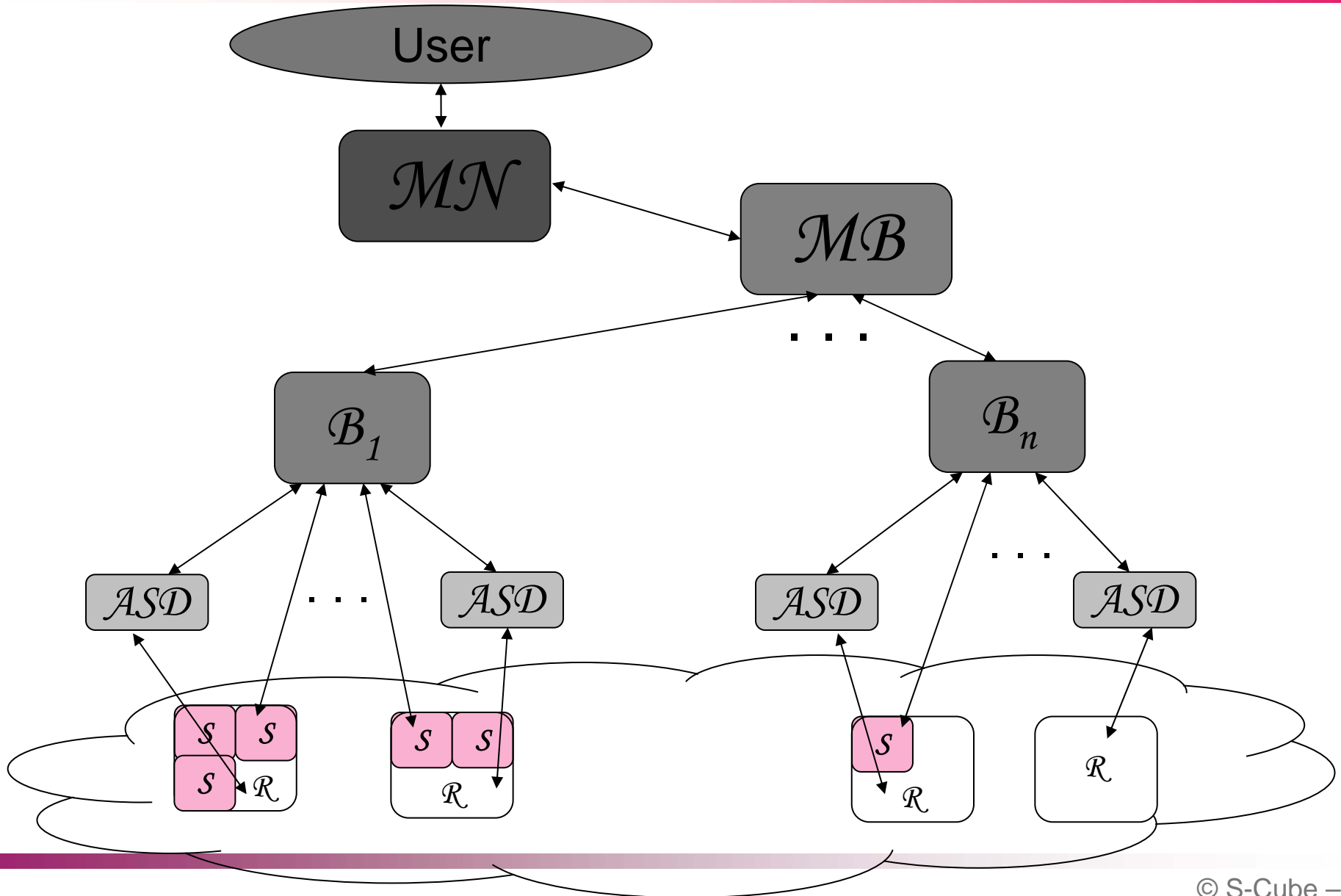


WP-JRA-2.3: Self-* Service Infrastructure and Service Discovery Support



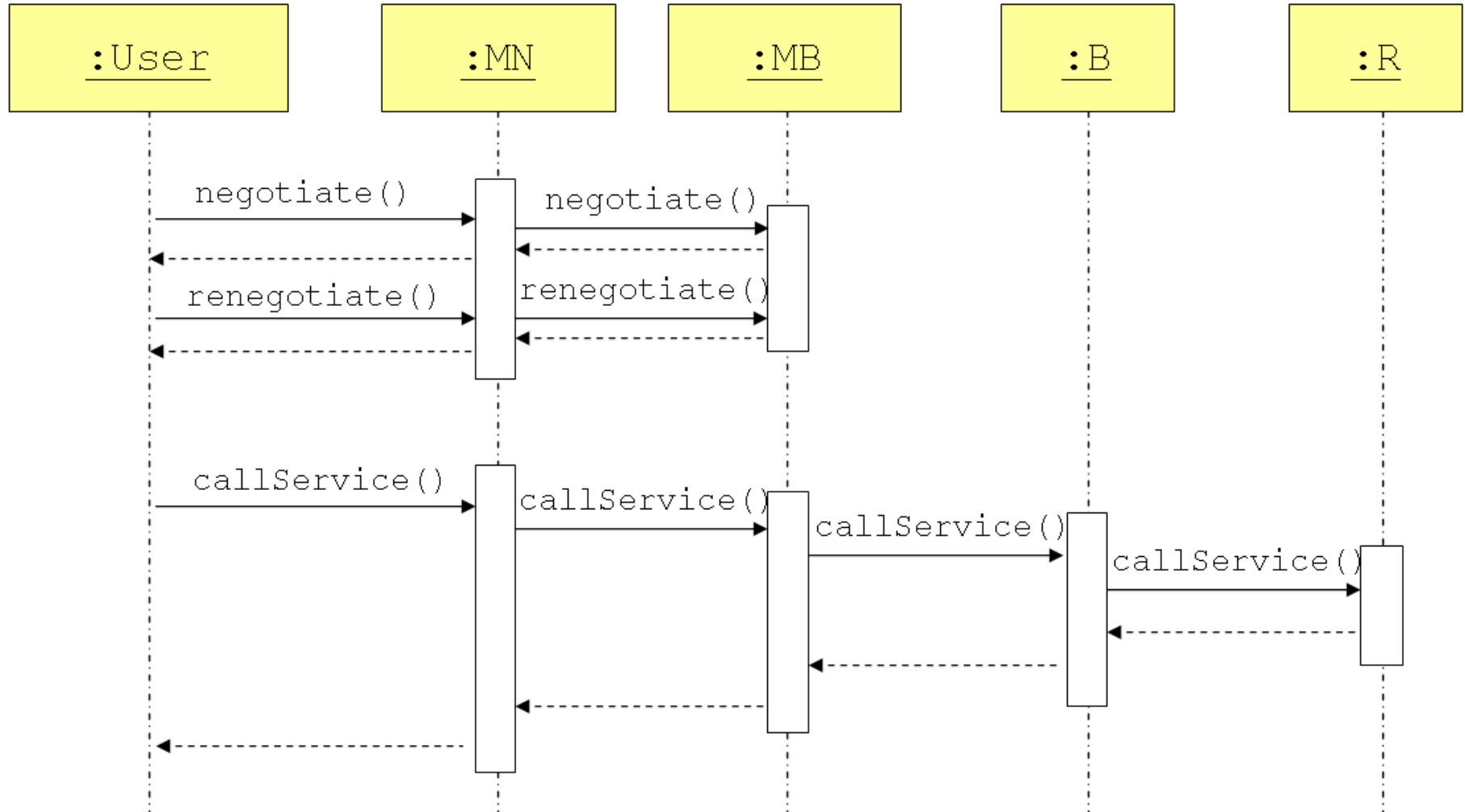


Model for SLA-based resource virtualization

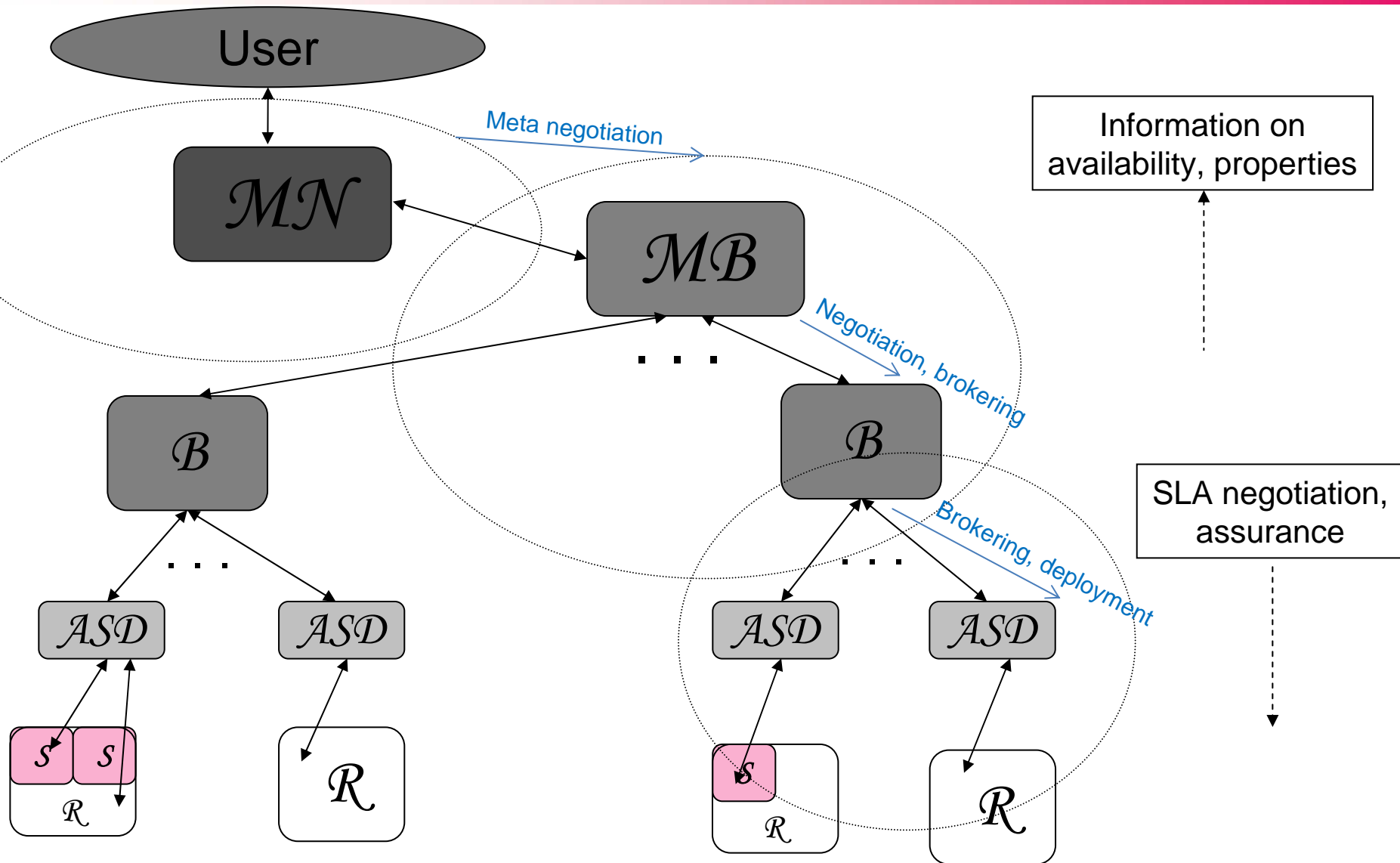


- User: A person, who wants to use a service
- MN – Meta-Negotiator: A component/service that manages Service-level agreements. It mediates between the user and the Meta-Broker, selects appropriate protocols for agreements; negotiates SLA creation, handles fulfillment and violation.
- MB – Meta-Broker: Its role is to select a broker that is capable of deploying/executing a service with the specified user requirements.
- B – Broker: It interacts with virtual or physical resources, and in case the required service needs to be deployed it interacts directly with the ASD.
- ASD – Automatic Service Deployment: It installs the required service on the selected resource.
- S – Service: The service that users want to deploy and/or execute.
- R – Resource: Physical machines, on which virtual machines can be deployed/installed.

Connections of the components of SRV

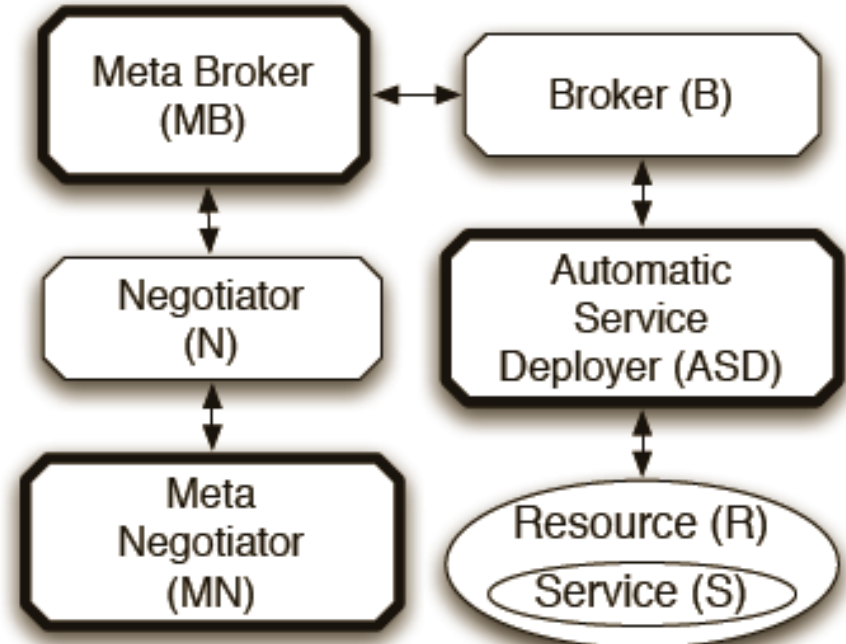
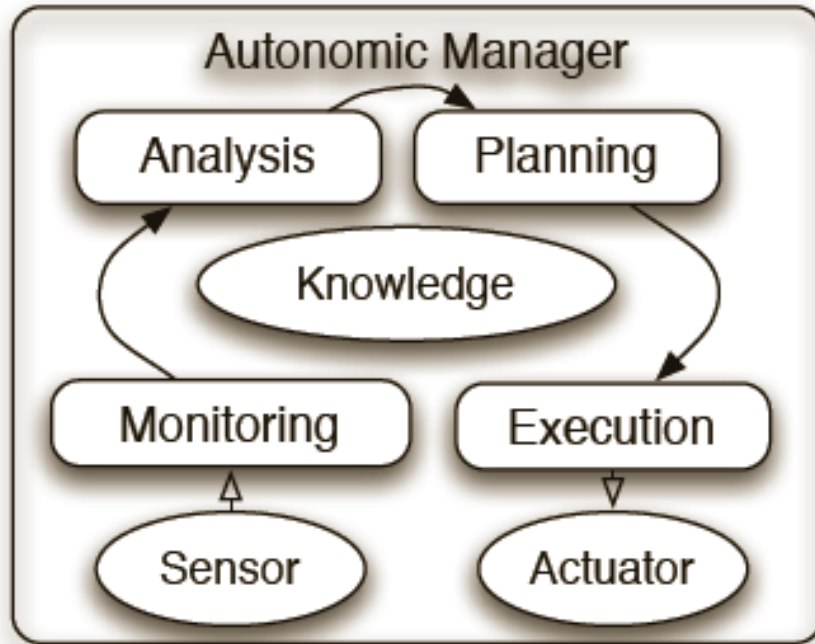


Target areas, operational steps



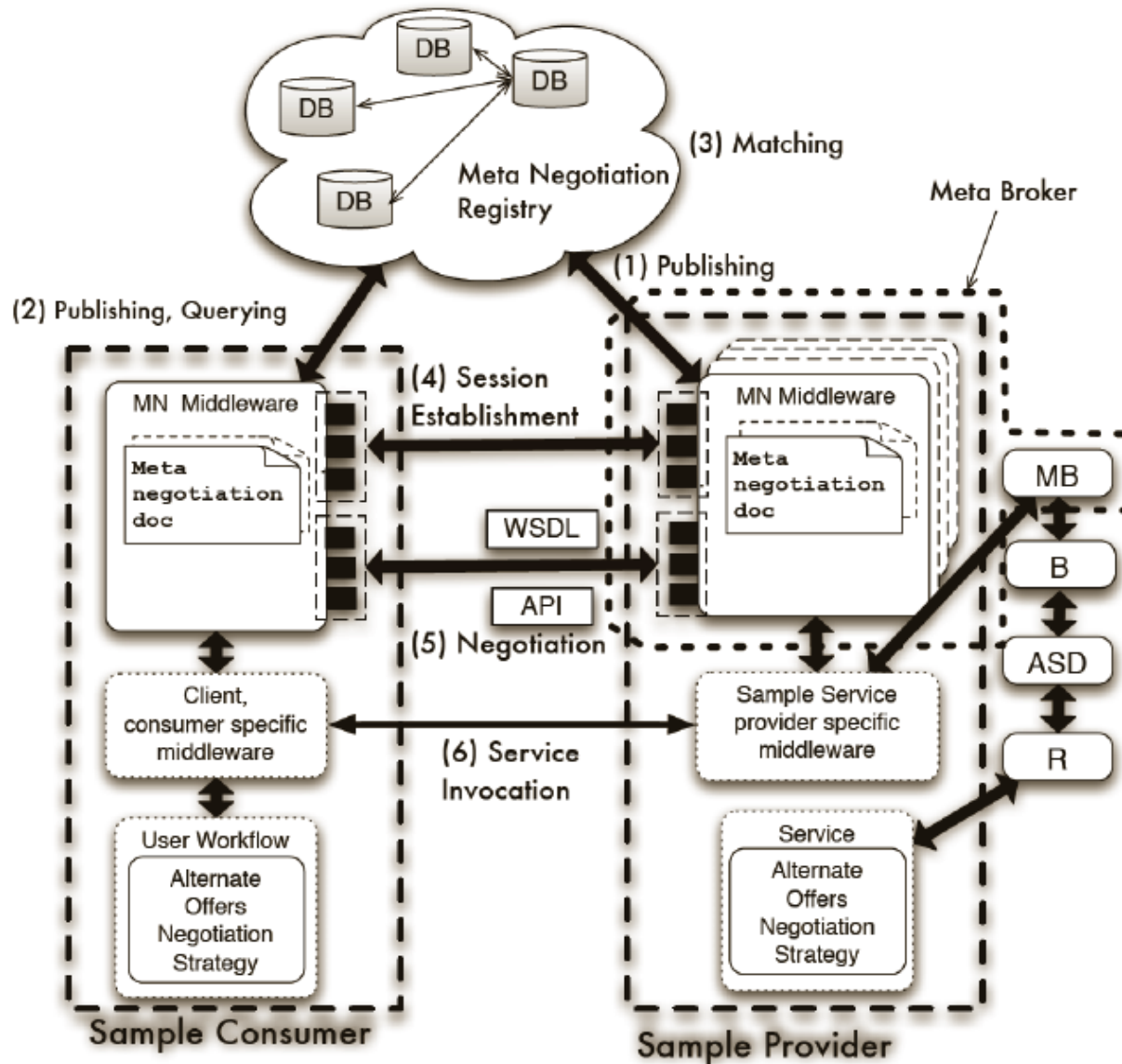
- User – MN: user supplies a specific meta-negotiation document
- MN – MB: agreeing on specific negotiation documents containing specific negotiation strategy to be used, negotiation protocols to be used (WSLA, WS-Ag,...) , terms of negotiation (e.g. time, price, ...), security infrastructure to be used
- MB – B: agreeing on a specific SLA written in a specific SLA language (e.g. WSLA, WS-Agreement) containing concrete SLA parameters like concrete execution time, concrete price, etc.
- B – ASD: agreeing on a specific service to be available on the ASD managed resources with the resource constraints resulted from the higher level negotiation – the service is going to be able to use the requested resources without disruptions from other parties
- Furthermore we need on each level (MN, MB, B, ASD) a negotiator which is responsible for generating and interpreting SLAs.

Autonomic components of the SRV architecture



Control loops of the autonomic components are implemented as **MAPE** (monitoring, analysis, planning, and execution) functions: The monitor collects state information and prepares it for the analysis. If deviations to the desired state are discovered during the analysis, the planner elaborates change plans, which are passed to the executor.

Meta-Negotiation in SRV



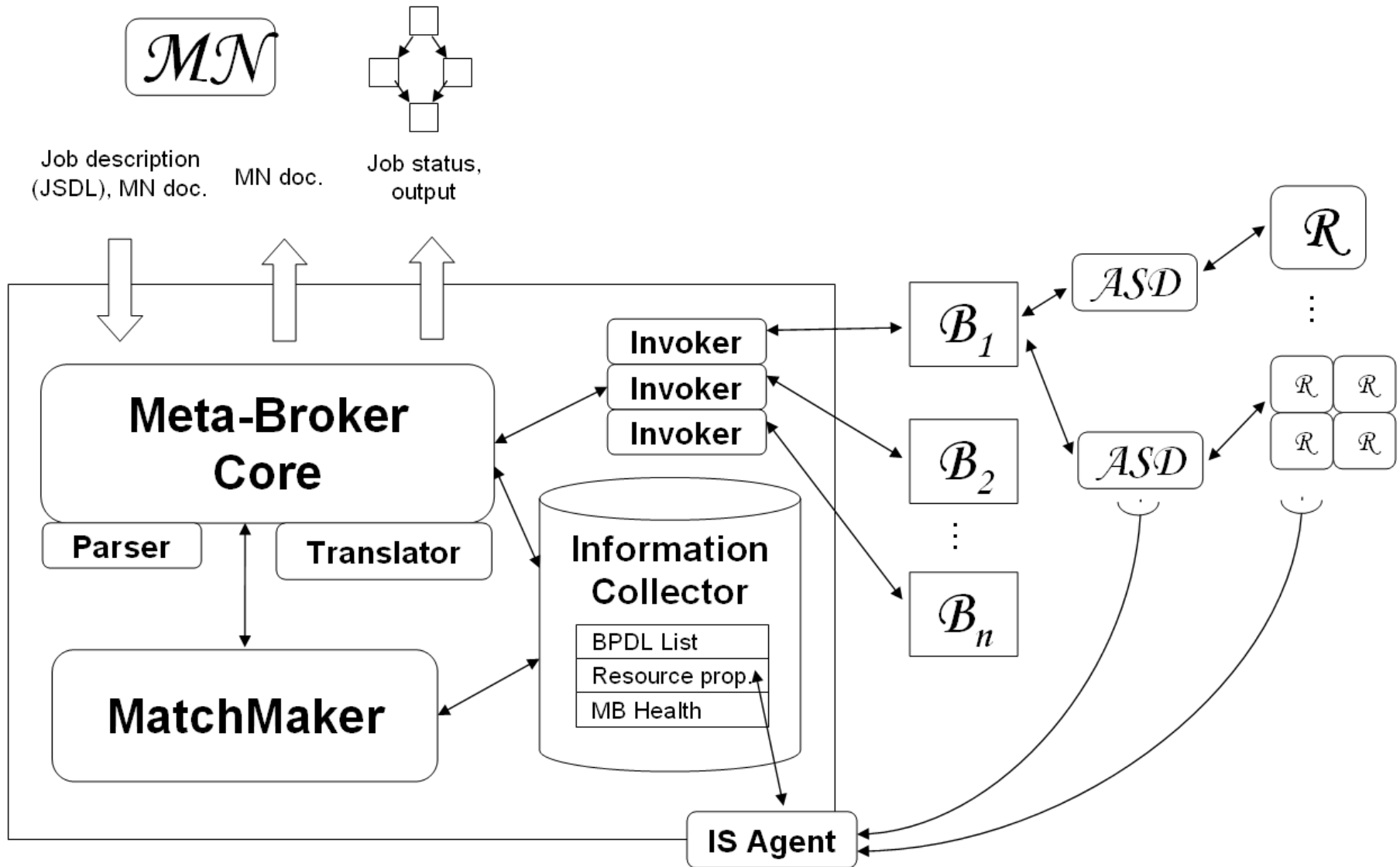
Sample Meta Negotiation document



```
<meta-negotiation xmlns:xsi=http://www.w3.org/2001/XMLSchema-instance ... >
  <entity> <ID name="1234"/> ... </entity>
  <pre-requisite>
    <role name="Consumer"/>
    <security>
      <authentication name="GSI"/><authorization name="xy"/>
    </security>
    <negotiation-terms>
      <negotiation-term name="beginTime"/> <negotiation-term name="endTime"/>
      <negotiation-term name="price"/>
    </negotiation-terms>
  </pre-requisite>
  <negotiation>
    <document name="WSLA" value="uri" version="1.0"/>
    <document name="WS-Agreements" value="uri" version="1.0"/>
    <protocol name="alternateOffers" schema="uri" version="1.0" location="uri"/>
  </negotiation>
  <agreement>
    <confirmation name="confirmator" value="arbitrator"/>
  </agreement>
</meta-negotiation>
```

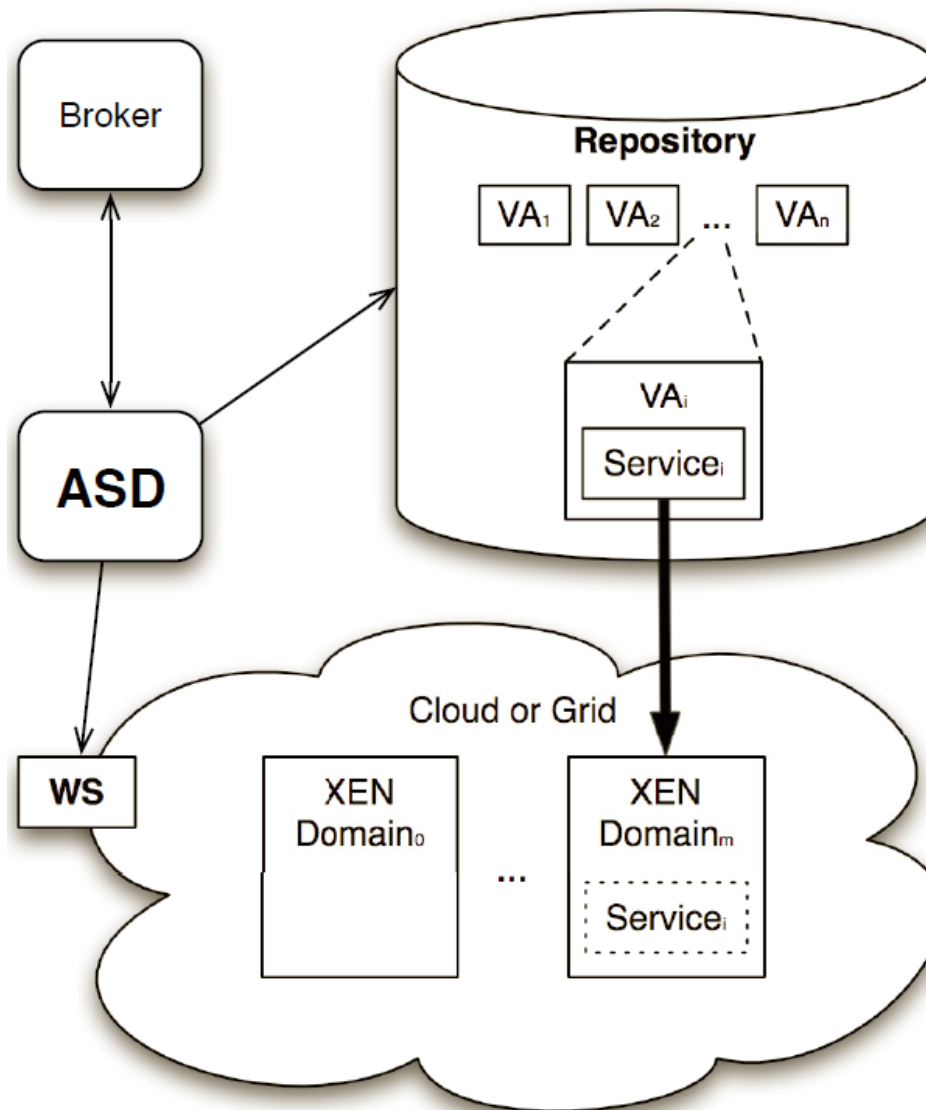
- **Publish.** A service provider **publishes descriptions** and conditions of supported negotiation protocols into the registry.
- **Lookup.** Service consumers perform lookup on the registry database by submitting their own documents describing the **negotiations** that they are looking for.
- **Match.** The registry discovers service providers who **support** the negotiation processes that a consumer is interested in and returns the documents published by the service providers.
- **Negotiate.** Finally, after an appropriate service provider and a negotiation protocol is selected by a consumer using his/her private **selection strategy**, negotiations between them may start according to the **conditions specified** on the providers document.
- The participants publishing into the registry follow a common document structure (ie. **meta-negotiation document**) that makes it easy to discover matching documents.

Meta-brokering in SRV



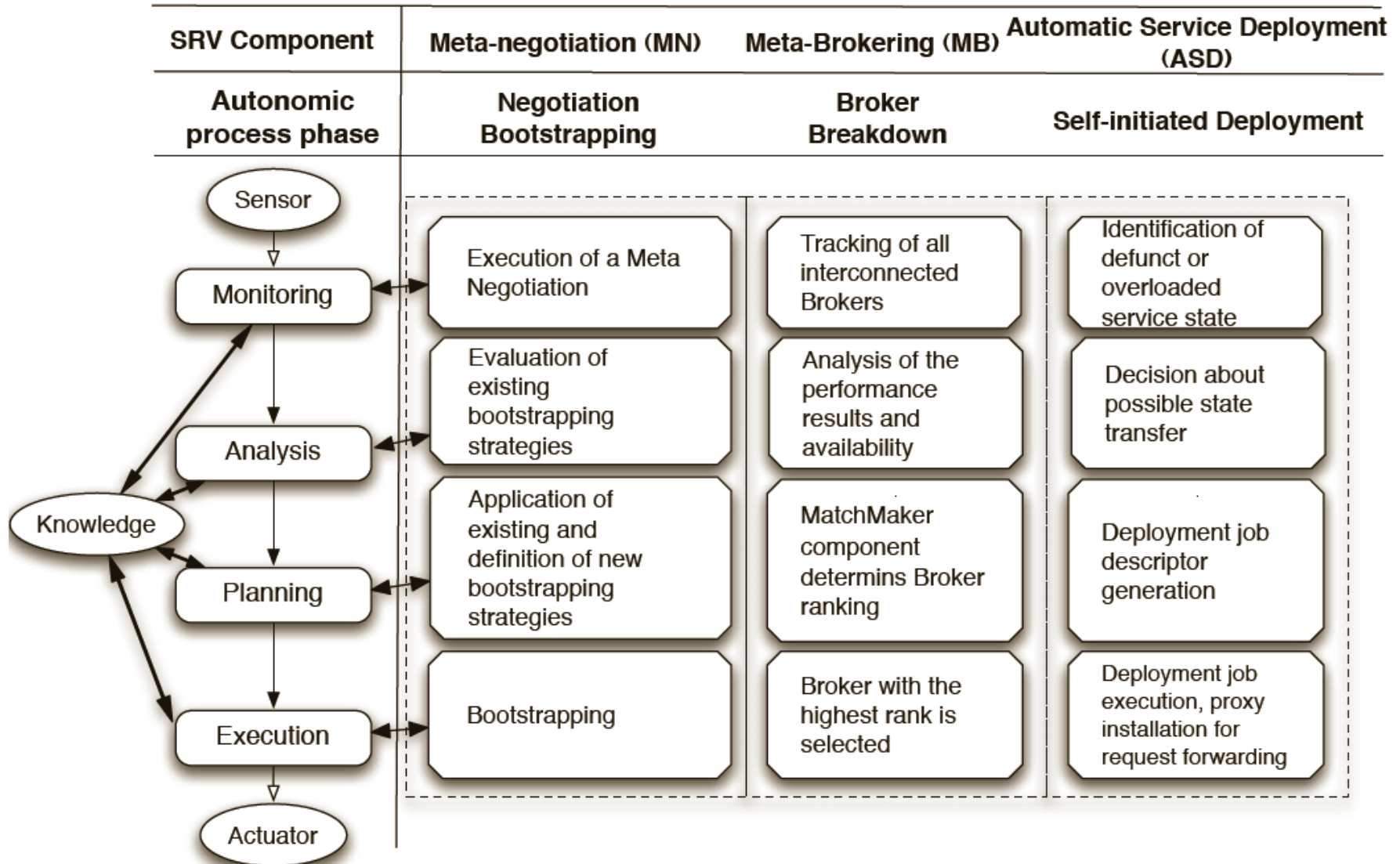
- The *Meta-Broker* is the core component: this communicates with the other components
- The *Translators* are responsible for transforming the user request to the language of the actually selected Broker (JSDL \leftrightarrow JDL, RSL, xRSL...)
- The *Invokers* hand over the job to the brokers and wait for the results, and provide additional information for the Information Collector about the submissions
- The *Information Collector* stores the connected broker properties and historical data of the previous submissions
- The *Matchmaker* compares the JSDL of the actual job to the BPDFL of the registered resource brokers, and selects a 'good' broker for the job (or service)
- The *IS Agent* regularly updates current properties and availability of the virtual resources reachable by the utilized brokers

Automatic Service Deployment in SRV

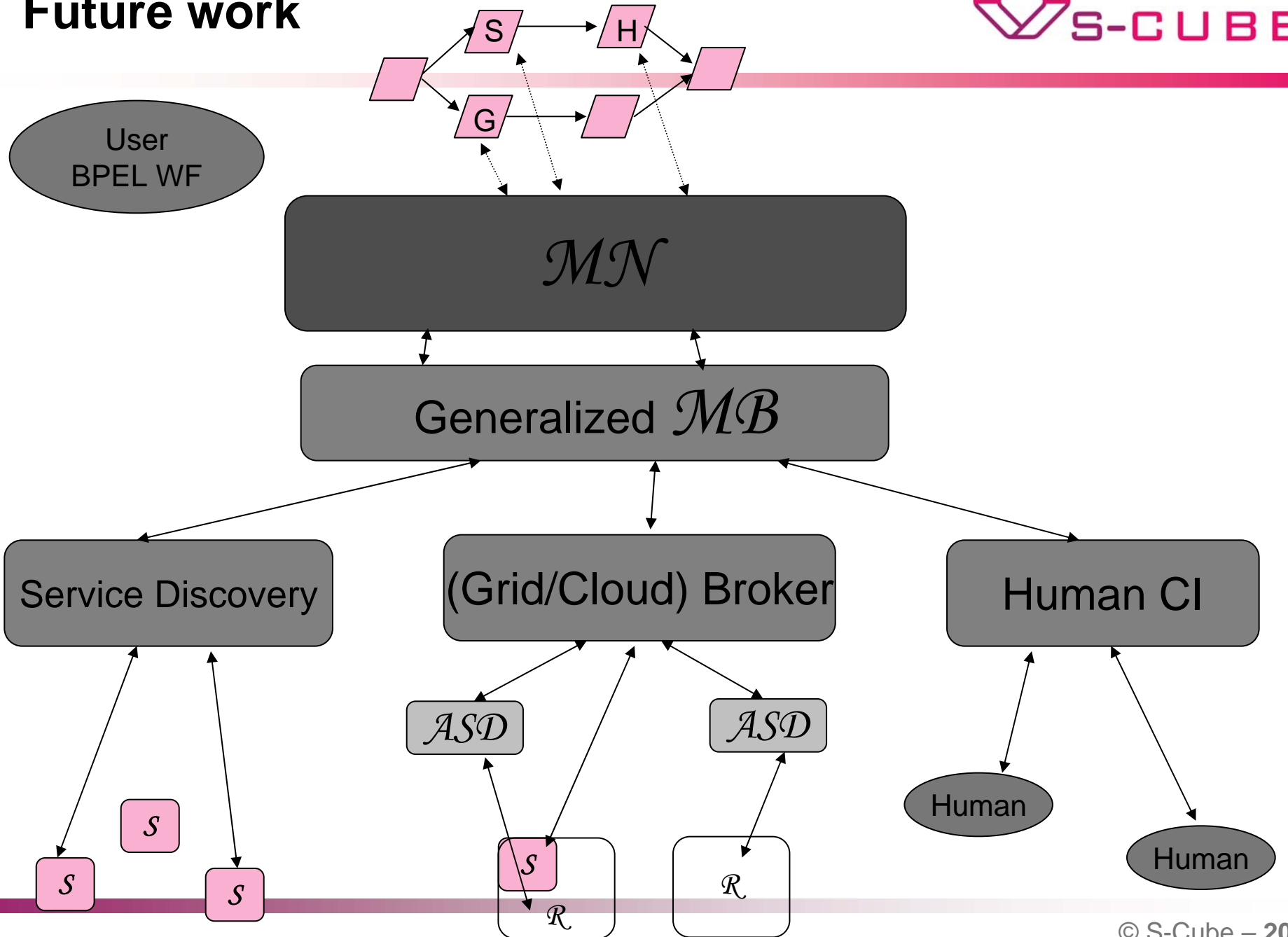


- **Repository** – holds the images of various services as ready to use virtual machine images (Virtual Appliances)
- **ASD** – Automatic Service Deployment coordinates the proper resource allocation for the given service according to the requirements from the broker
- **WS** – Workspace service, offers the virtualization capabilities – virtual machine creation, removal and management - of a given grid/cloud site as a WSRF service

Self-management examples in the SRV



Future work



Conclusions

- The presented autonomic SLA-based resource virtualization with on-demand service deployment incorporates **three** enhancements:
 - a meta-negotiation component for generic SLA management
 - a meta-brokering component for diverse broker management
 - and an automatic service deployment for resource virtualization on the Cloud
- We have shown, how the principles of autonomic computing can be incorporated to the SRV architecture, and demonstrated **MAPE** actions with case studies.
- Our **future work** aims at finalizing the presented components and interfacing the architecture to commercial Clouds and production Grids.