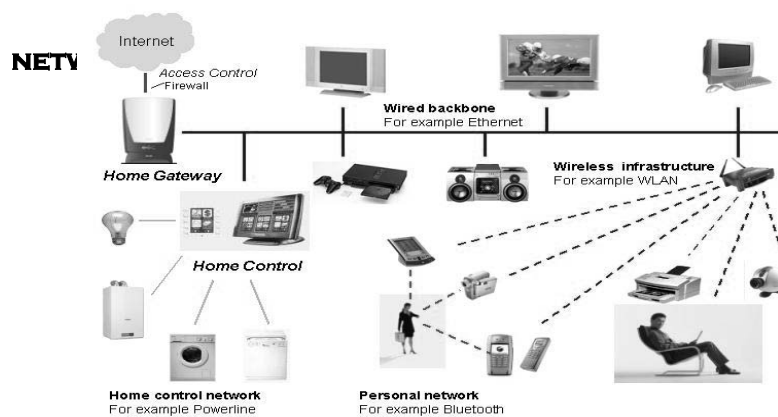


## Z25 Adaptive and Mobile Systems Dr. Cecilia Mascolo

## INDISS: Interoperable Discovery System for Networked Services

Yérom-David Bromberg and Valérie Issarny  
INRIA-Rocquencourt

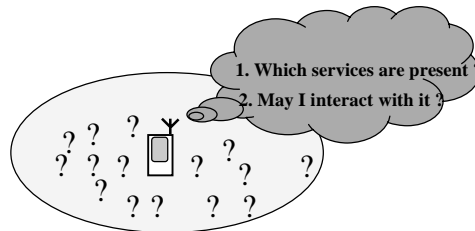
### TODAY'S OPEN



=> Middleware heterogeneity

## MIDDLEWARE HETEROGENEITY

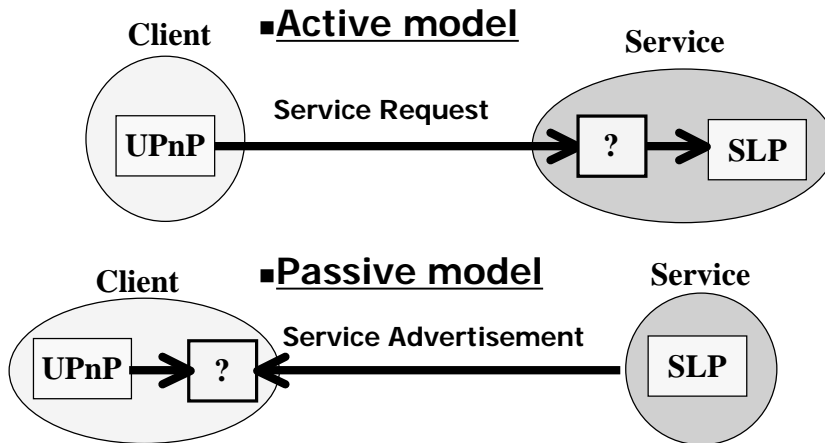
- **Two base middleware functions for pervasive computing**
  - **Service discovery (SDP)**
  - Broker for communication between services



## Service Discovery Protocols

- There are many service discovery protocols (Jini, SLP, UPnP, etc)
- Some rely on a central repository (which could however be dynamic, ie change identity in time)
- Some do not rely on central repository (nodes discover each others)
  
- Passive discovery: client listens for advertisements
- Active discovery: client sends probes asking for info

SDP HETEROGENEITY

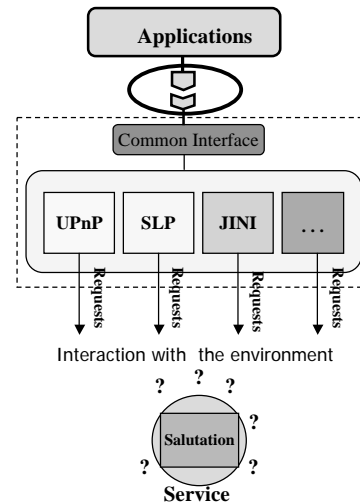


SOLUTIONS TO INTEROPERABILITY

- **Reflective middleware**
  - E.g., ReMMoC at U. Lancaster
- **Software architectures**
  - Component & connector abstractions for system modeling

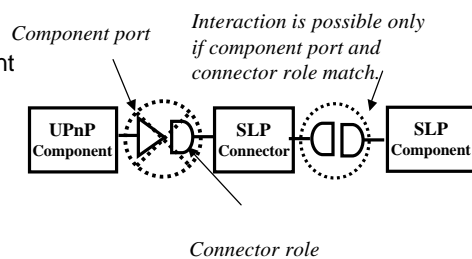
### REMMoC

- **ReMMoC**  
[Grace, Blair & Samuel, ICDCS – MCM'03]
  - Dynamic integration of SDP-specific “plug-ins”, according to context
- Quality of SDP detection dependent upon embedded plug-ins
- High latency
- Significant resource consumption (bandwidth, memory & CPU)
- Introduces specific/proprietary SDP for client applications



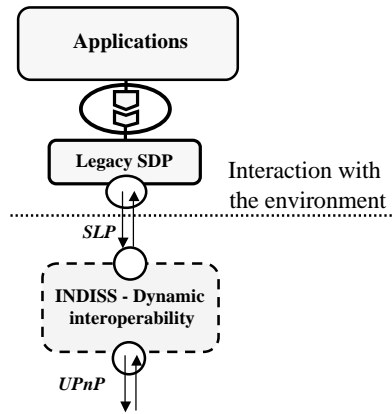
### SOFTWARE ARCHITECTURES (SA)

- Decouple applications from interaction protocols / middleware
- SA & interoperability
  - Dynamic matching of component and connector through adaptation of port & role behavior
- Events-based interoperability
  - Inspired by SA concepts [Ryan & Wolf, ICSE'04]
- Events to solve syntactic and semantic mismatches among SDPs

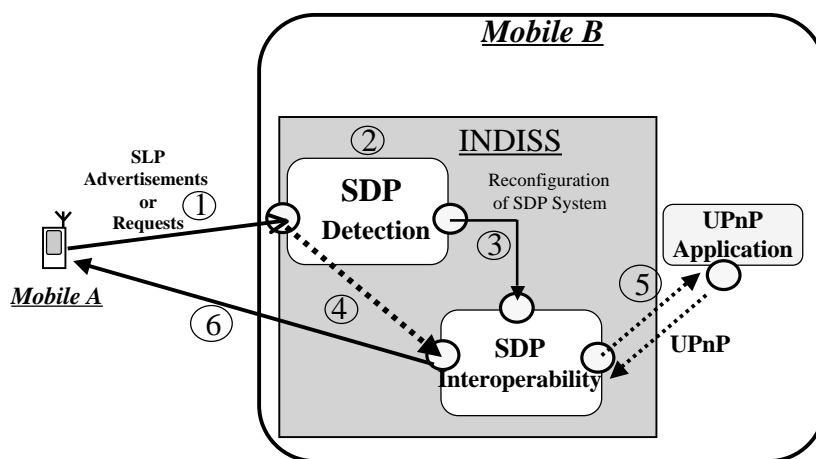


**INDISS: AN OPEN SOLUTION TO SDP INTEROPERABILITY**

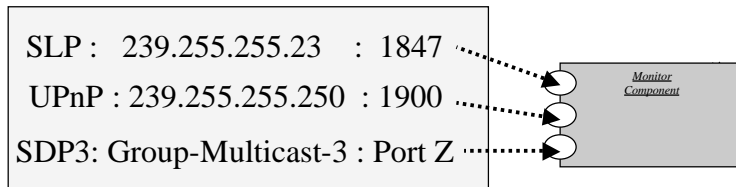
- Application services developed using existing middleware
- Middleware-layer interoperability that is transparent to applications
- Event-based interoperability



**THE INDISS SYSTEM**

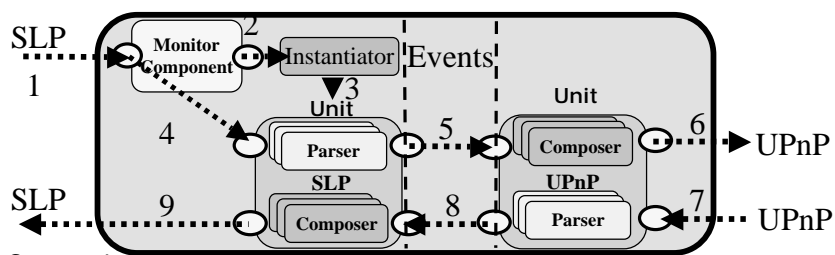


SDP DETECTION



- **The INDISS monitor component**
  - Maintains the table of identification tags
  - Listens on all the ports provided by the table
- Null cost in terms of bandwidth consumption as the monitor passively listens on SDP ports

SDP INTEROPERABILITY

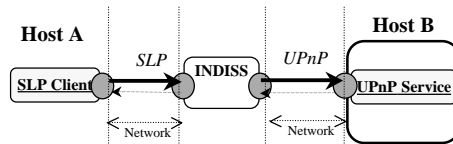


- Communication between *parser* & *composer*.
  - Independent of the SDPs' syntactic details
  - Done at the semantic level using events
  - There exists a set of events common to all SDPs
    - Control events (system configuration), network events (network-level messaging), service events (service discovery), request events (request content), response events (response content)

### Example SLP to/from UPnP

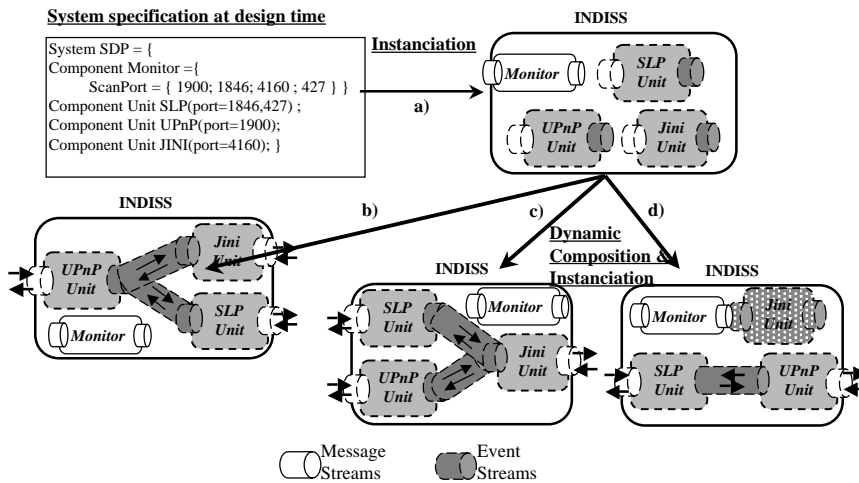
- Monitor forwards an SLP req to SLP parser
- Parser generates a stream of events for UPnP composer

### EXAMPLE OF TRANSLATION FROM SLP TO UPnP



Request	Generated Events	Composed request
SLP Search	<p>SDP_C_START ....  <b>SDP_NET_MULTICAST</b>  SDP_NET_SOURCE_ADDR  <b>SDP_SERVICE_REQUEST</b>  SDP_REQ_VERSION  SDP_REQ_SCOPE  SDP_REQ_PREDICATE  SDP_REQ_ID  <b>SDP_SERVICE_TYPE:</b>  SDP_C_STOP</p>	<p><i>From the previous events the UPnP unit multicast a UPnP search request to discover UPnP services in its vicinity:</i></p> <p>M-SEARCH * HTTP/1.1  SERVER: 239.255.255.250:1900  ST: urn:schemas-upnp-org:device:clock  MAN: ssdp:discover  MX: 0</p>

## INDISS CONFIGURABILITY

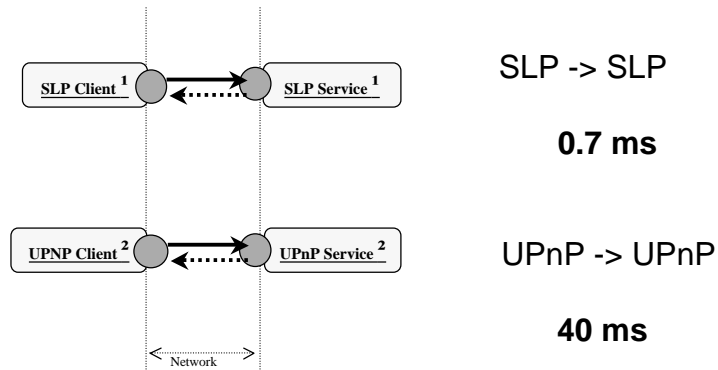


## Overhead and Size

- SLP and UPnP interoperability without INDISS 514KB
- UPnP client +INDISS 598KB
- SLP client +INDISS 352KB
- However when more services are added 2 new libraries are added if INDISS is not used.

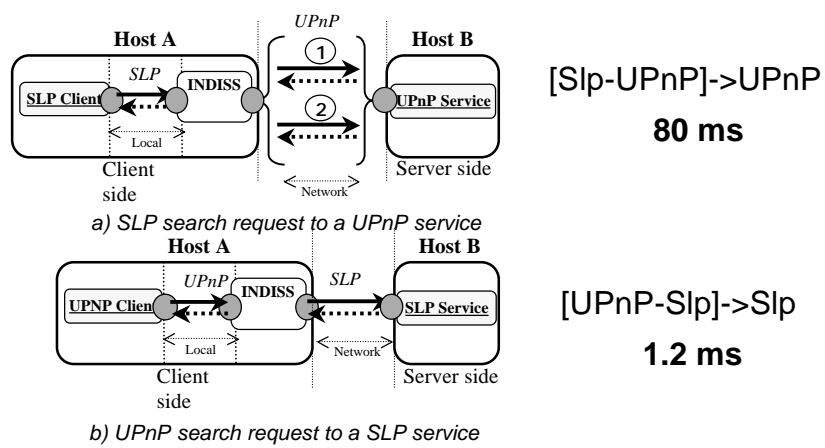


EVALUATION – NATIVE CASE



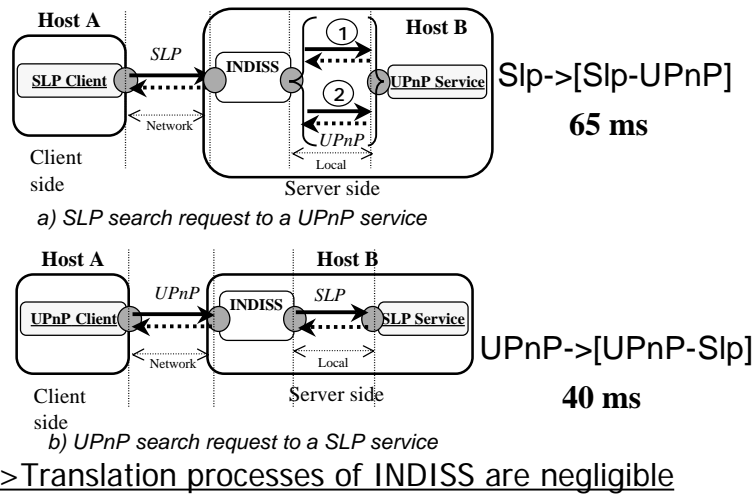
<sup>1</sup>Java Open Slp  
<sup>2</sup>Cyberlink UPnP Stack

EVALUATION – ON CLIENT



=> Translation processes of INDISS are negligible

EVALUATION – ON SERVER



CONCLUSION

- **An approach to interoperability**
  - Transparent interoperability
  - Dynamic translation of one SDP to another
  - Solution based on service-oriented architecture
  - Interoperable service discovery at the network layer
- Questions about interoperability in other cases...