An Open Source H.323 / SIP Gateway as Basis for Supplementary Service Interworking

Ralf Ackermann, Vasilios Darlagiannis, Manuel Goertz, Martin Karsten, Ralf Steinmetz

Darmstadt University of Technology
Industrial Process and System Communications (KOM)

Merckstr. 25, D-64283 Darmstadt

{Ralf.Ackermann, Vasilios.Darlagiannis, Manuel.Goertz, Martin.Karsten, Ralf.Steinmetz}@KOM.tu-darmstadt.de

Outline

http://www.kom.e-technik.tu-darmstadt.de

Starting Situation and Intentions

Challenges:

- **co-existence of at least H.323 and SIP for IP-Telephony**

- **ongoing development and starting deployment with**
  - various building blocks from different developers and vendors
  - demand for interworking in various different scenarios

- **gateways (signaling proxies, translators) as a general trend**

- **robustness and carrier-grade services needs a more formalized development process**

- $\Rightarrow$ **need for a framework for both**
  - rapid prototyping of components and services
  - verification testbed
  - deployment

Intentions

- **industry cooperation targeting at practically usable solution**
  - on top of state-of-the art mechanisms and components
  - benefit from the work of others
    - Kundan Singh / Henning Schulzrinne - last years paper and draft

- **integrate different (emerging) H.323 and SIP stacks**

- **stay open for integrating new or enhanced components / services**
  - high dynamics in development
  - different grade of maturity

- **consider providing gateways for resource-bound devices - PDAs, Networked Appliances ...**
  - theory (and practical experiments) show that H.323 is just to expensive
  - ohphone using about 9 MByte memory footprint on a PDA device (Compaq iPAQ with 32 MB RAM, 16/32 MB Flash is just to much)
  - even C++ SIP applications are currently large
  - using very bare-bone / low-complexity signaling as a future task

H.323 / SIP Interworking - Basic Requirements

Basically:

- **connecting RTP media endpoints for both H.323 and SIP-originated calls**
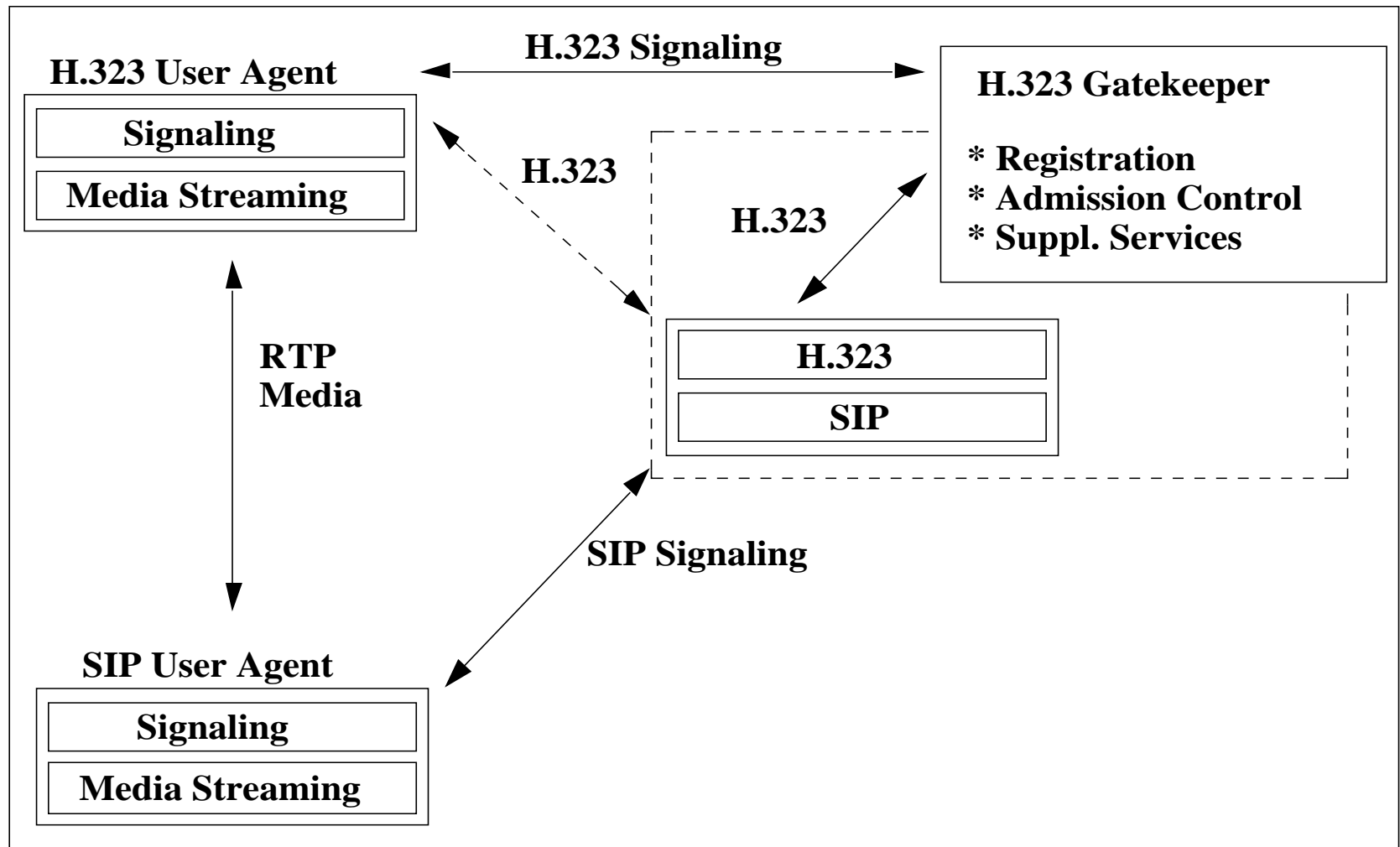
In-detail:

- **Mapping of protocol elements and sequences**
  - alerting, codec and endpoint negotiation, call teardown
  - this is not straight-forward especially due to different protocol semantics and various versions on H.323 side

- **Support for different "infrastructure integration" styles**
  - end system to end system
  - subscriber based
    - SIP-centric
    - H.323-centric
  - connecting "protocol clouds"

- **Support for different address mapping schemes**
  - gateway based
  - using interconnected protocol mechanisms (e.g. REGISTER) themselves

- **Scalability - Support for multiple calls at a time**

iptel2001.fm

Initial Implementation

H.323 User Agent

| Signaling |
| Media Streaming |

H.323 Signaling

H.323 Gatekeeper

* Registration
* Admission Control
* Suppl. Services

H.323

H.323

| H.323 |
| SIP |

RTP Media

SIP Signaling

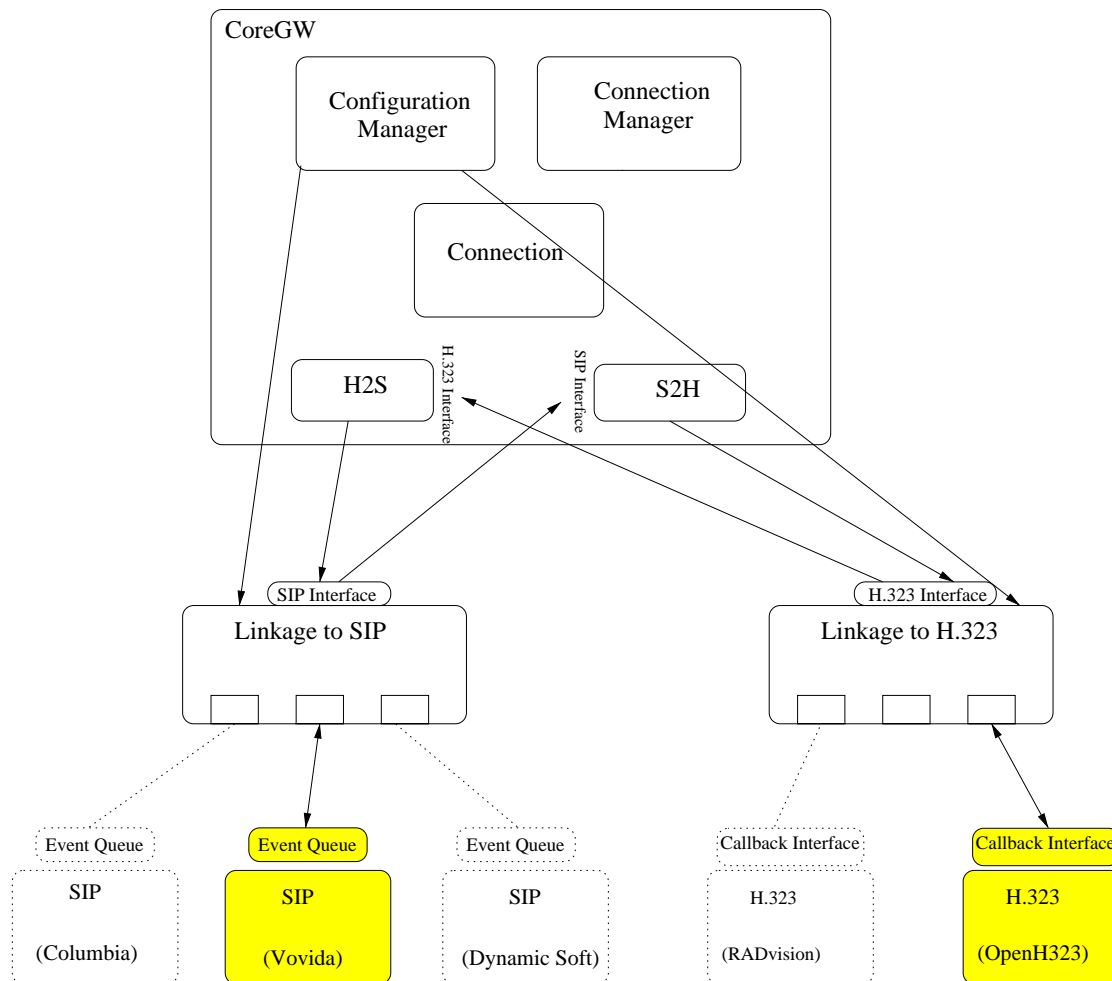**SIP User Agent**

| Signaling |
| Media Streaming |

- **stacks have been chosen after evaluating certain criteria**
  - see paper for evaluation list
  - OpenH323 / vovida SIP

Redesign using an abstraction layer

- **lack of a uniform and stable H.323 or SIP API**

- **interworking functionality inside a stable system core**

- **"linkage" to stacks**

- **OO-abstraction:**
  - connection(s)
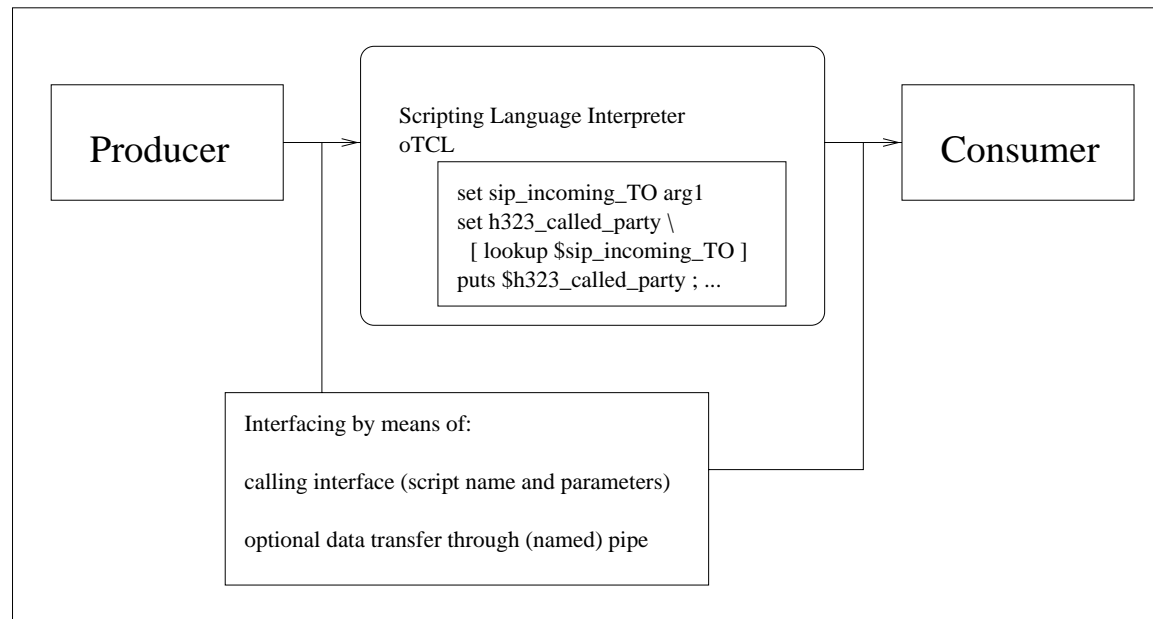 - instantiation of multiple threads

iptel2001.fm

Rapid Prototyping and Testing using Scripting

- **(x)oTcl approach:**
  - scripting languages (in contrast to system programming languages) as the key programming means for the 21st century - form the "glue"
  - allow for:
    - fast prototyping using run-time interpretation
    - dynamic extensibility (C-linkage using shared libraries)
    - extensions add object-orientation

- **initially used for simple tasks like comfortable address mapping**

Producer → Scripting Language Interpreter oTCL

```
set sip_incoming_TO arg1
set h323_called_party \
  [ lookup $sip_incoming_TO ]
puts $h323_called_party ; ...
```

→ Consumer

Interfacing by means of:

calling interface (script name and parameters)

optional data transfer through (named) pipe

- **adapted for more general tasks (FSM states and operators)**
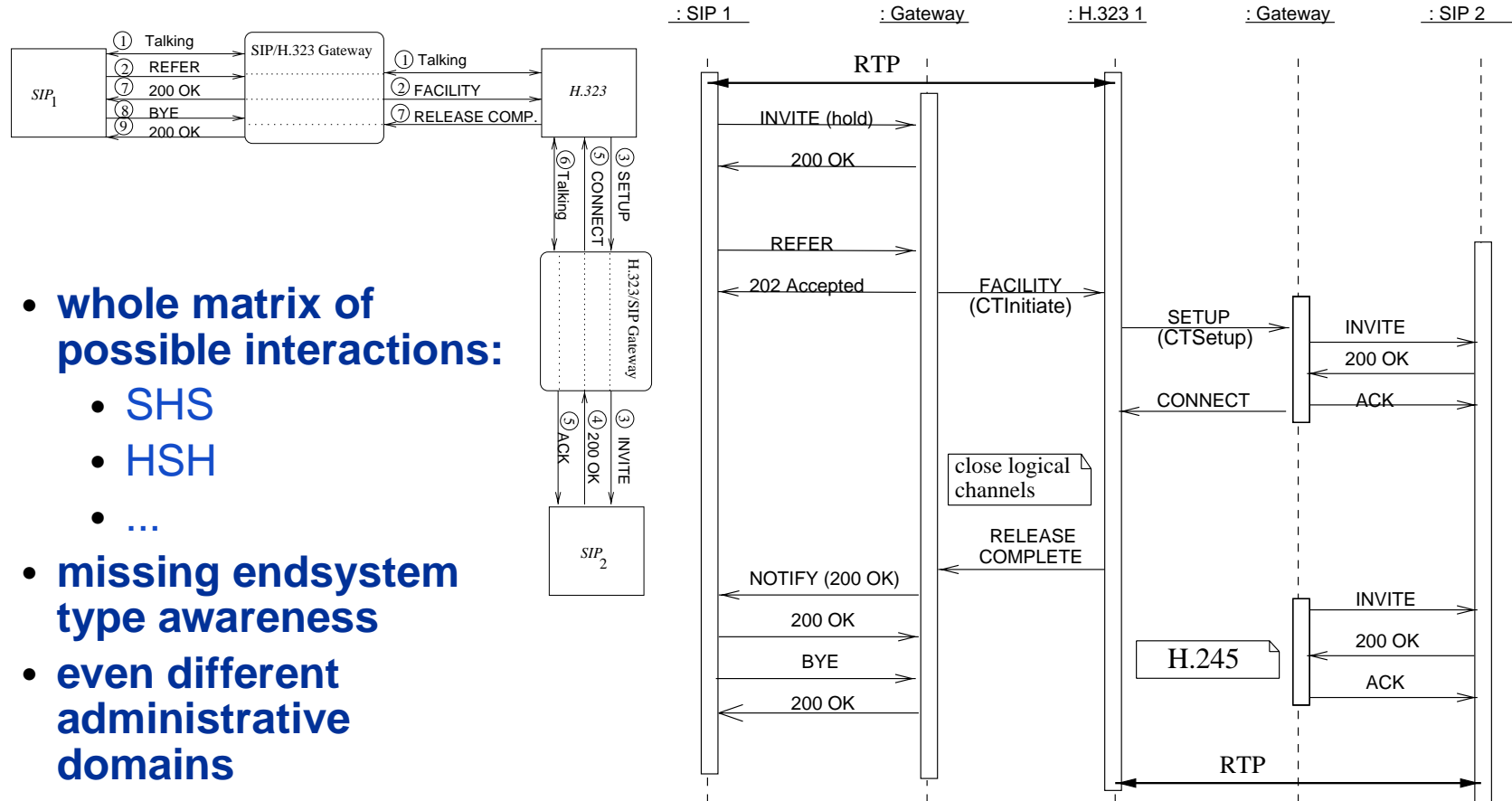
Supplementary Service Interworking

Connecting Media Streams is comparable straight forward, whereas providing "services" is THE major challenge

- both "Supplementary" as well as "Value Added" Services

- **ITU H.450.x - we concentrated on a subset first**
  - H.450.1 - Framework
  - H.450.2 - Call Transfer
  - H.450.3 - Call Diversion
  - ...

- **in SIP - we have concepts**
  - Lennox / Schulzrinne / La Porta "Implementing Intelligent Network Services with SIP"
  - description of implementation mechanisms regarding the "Value Added Services"
    - CPL, SIP CGI, SIP Servlets, implementation A, B, C ...
  - description of protocol mechanisms (targeted at H.450.x like Services)
    - Call Control Framework

- **in general - less strict and determined**
  - interactions make system approach desirable and even necessary

Supplementary Service Interworking - Concepts

- **Unattended / Blind Call Transfer (SIP1 - H.323 1 => SIP 2 - H323 1)**

  - **whole matrix of possible interactions:**
    - SHS
    - HSH
    - ...
  - **missing endsystem type awareness**
  - **even different administrative domains**

- **individual components exist and can be tested**
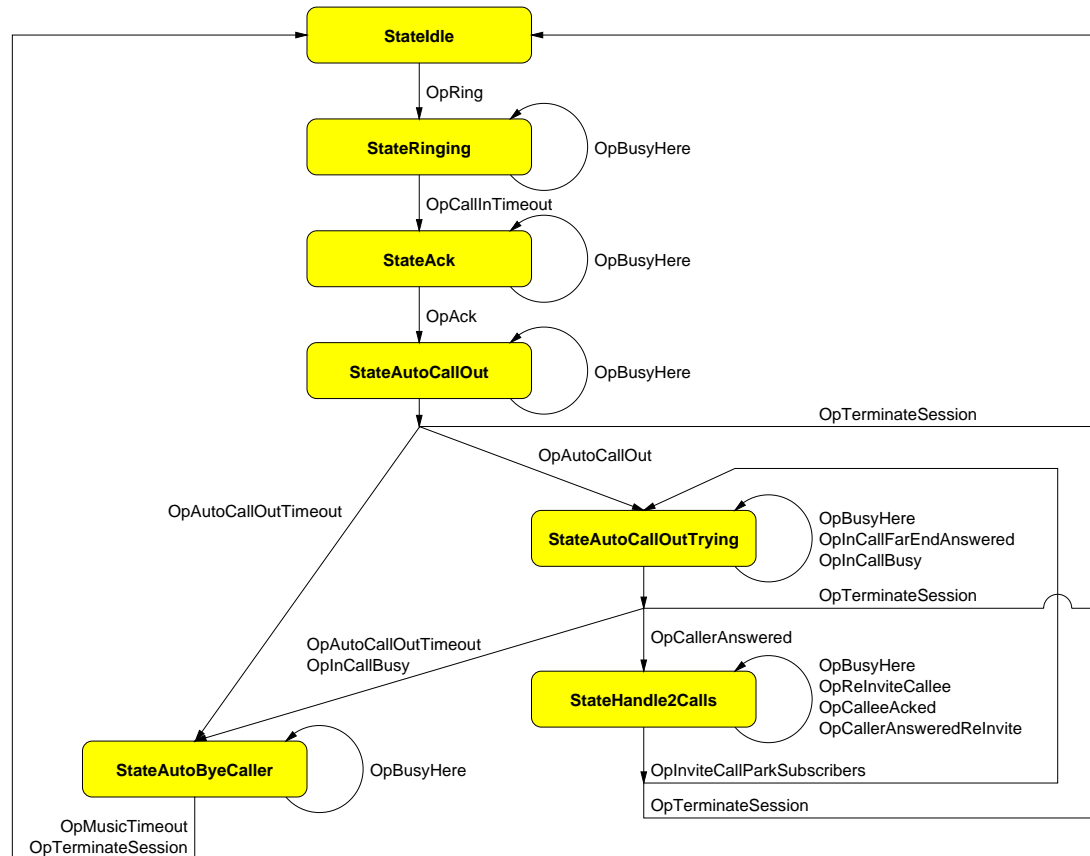- **integration approach must show to work in practice**

Supplementary Services Interworking - SIP side

- **implemented and tested**
  - benefiting already from the easy FSM extensibility
  - Call Transfer (INVITE / RE-INVITE based - further work using REFER)
  - Call Park and Pickup using an additional Call Park Server and an enhanced Vovida sua SIP user agent

- **FSM directly derived from message sequence diagram**

- **H.450 integration currently under investigation**

Conclusion and Future Work

Starting point:

- **Industry Project with straight-forward task to fulfill**

Results:

- **working solution based on Open Source**
- **Linux as suitable platform with results valuable for transition to other systems (e.g. VxWorks)**
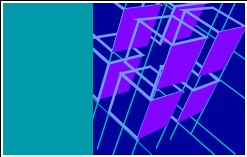- **contribution: more general framework for enhancement**

Future Work:

- **another straight-forward task (make H.450 / SIP interaction real)**
- **testbed for applying and testing formal approaches**

References and related work:

- **Singh / Schulzrinne - siph323**
- **Vovida VOCAL package implementing stacks and protocol translators (OpenH323, vovida SIP) now**

Thank you !

Questions, Comments ?