TeliaSonera

IN Functions with Call Processing Language and Common Gateway Interface

Jaakko Vuorensola
Thesis Seminar on Networking Technology
Helsinki University of Technology

Basic information

- Thesis written at Sonera Carrier Networks Ltd.
- Supervisor: prof. Jouni Karvo (pro tem)
- Instructor: Kari Eggen (M.Sc.)
- Thesis started in August 2004
- Literature study that was further applied to existing IN functions

Contents

- Background information
- Objectives of the thesis
- Traditional IN concept
- Service creation with SIP
- CPL
- CGI for SIP
- Service triggering and charging
- Applied IN functions
- Conclusions

Background

- Migration to IP technologies
  - PSTN switches replaced with Media Gateways and Media Gateway Controllers
  - Traditional trunking signalling replaces with IP-based signalling protocols
  - 3GPP and ETSI have chosen SIP for the signalling protocol in 3G and NGN
- At the moment intelligent voice services are offered from the CS domain also for SIP subscribers
  - Limited gateway resources
  - What if the same services are offered directly from the PS domain?

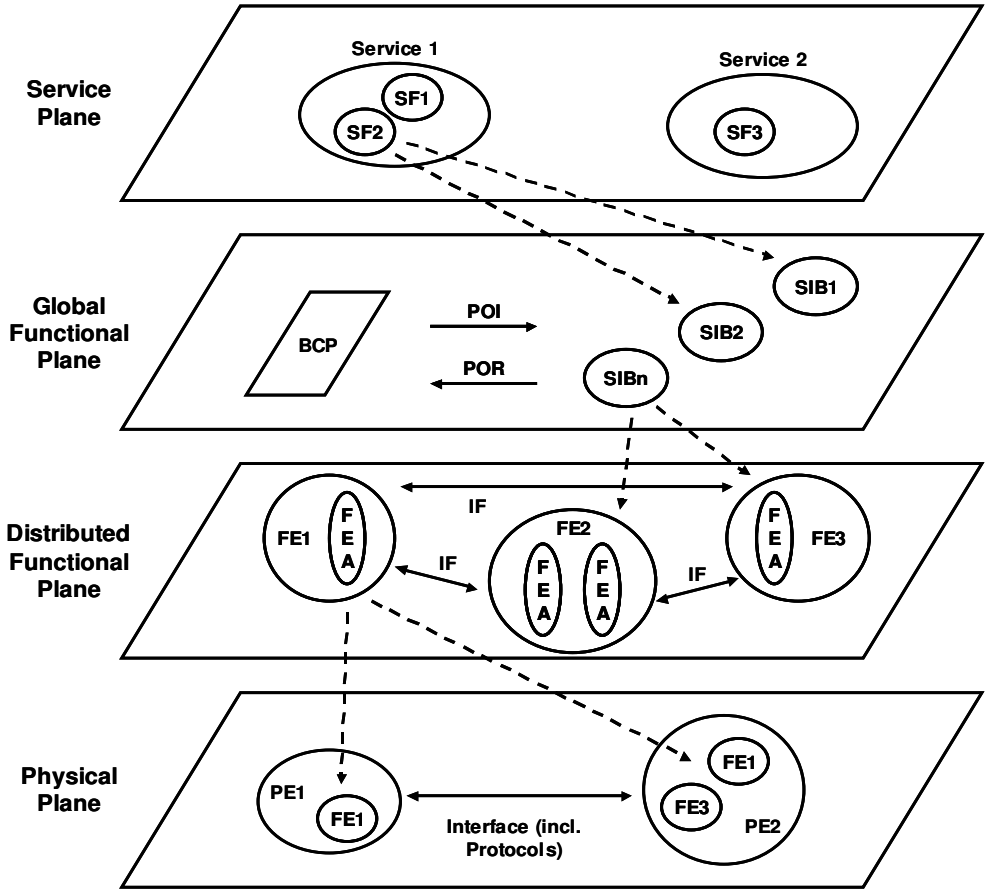
Objectives of the thesis

- Evaluate if the IN concept can meet the requirements of VoIP services
- Introduce Call Processing Language (CPL)
- Introduce Common Gateway Interface for SIP (SIP-CGI)
- Evaluate how CPL and SIP-CGI affect the service creation process
- Evaluate the service creation possibilities of CPL and SIP-CGI in the scope of current IN functions

Traditional IN concept

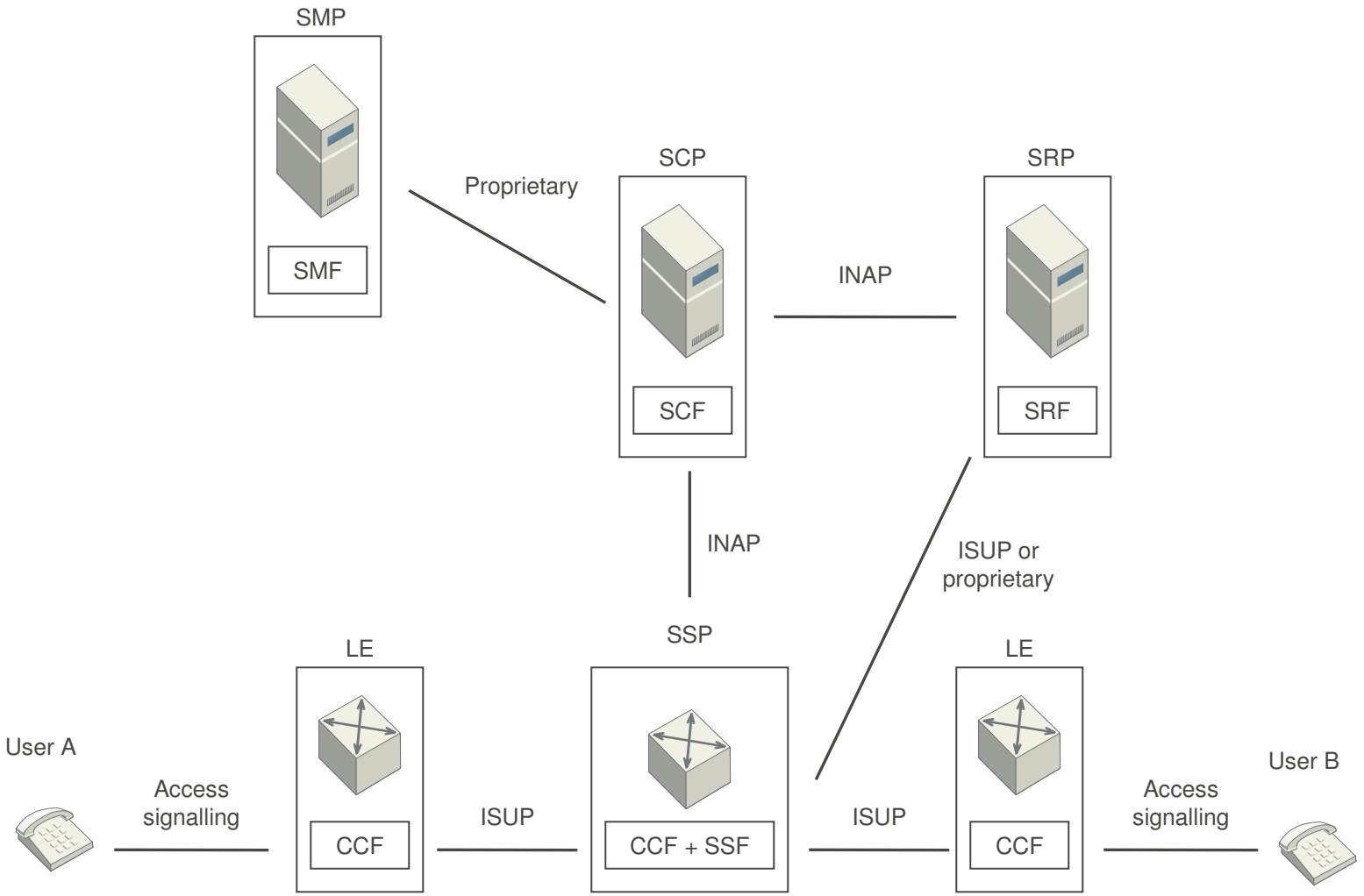
- Standardized by ITU-T and ETSI
- Standardization started already in the beginning of 1990s
- Standardization carried out in phases called Capability Sets (CS)
  - At the moment CS-1 – CS-4 have been defined
  - Most of the IN services are still based on CS-1
  - The deployment of CS-3 and CS-4 has been rare, if any
- Services are modelled with IN Conceptual Model, which provides four different abstraction levels to the service modelling
  - Higher levels of abstractions offer easier service creation but the possibilities are more restricted
- INAP protocol between different IN elements

IN Conceptual Model



SF	Service feature	IF	Information flow
BCP	Basic call process	FE	Functional entity
POI	Point of initiation	FEA	Functional entity action
POR	Point of return	PE	Physical entity
SIB	Service independent block		

IN Physical Entities

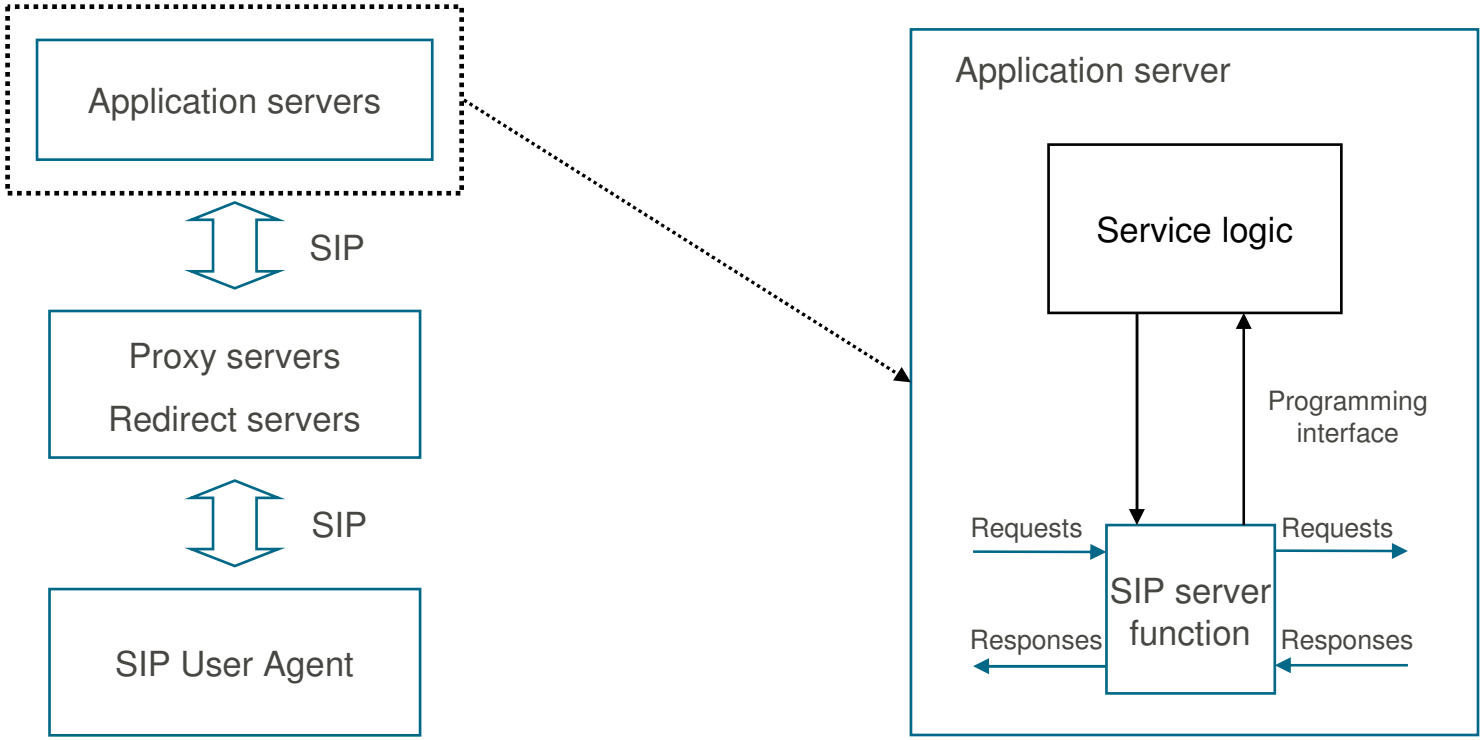


IN shortcomings

- The call model is geared towards conversational services only
  - How to control multimedia sessions?
- A lot of vendor specific solutions on the market
  - Proprietary service creation environments
  - Proprietary INAP versions
- As a result
  - Services are not directly portable between different systems
  - Steeper learning curves
  - Smaller amount of developers
  - Longer development times

Service creation with SIP

- Service logic, which controls proxy and redirect behaviour, is added to a SIP server

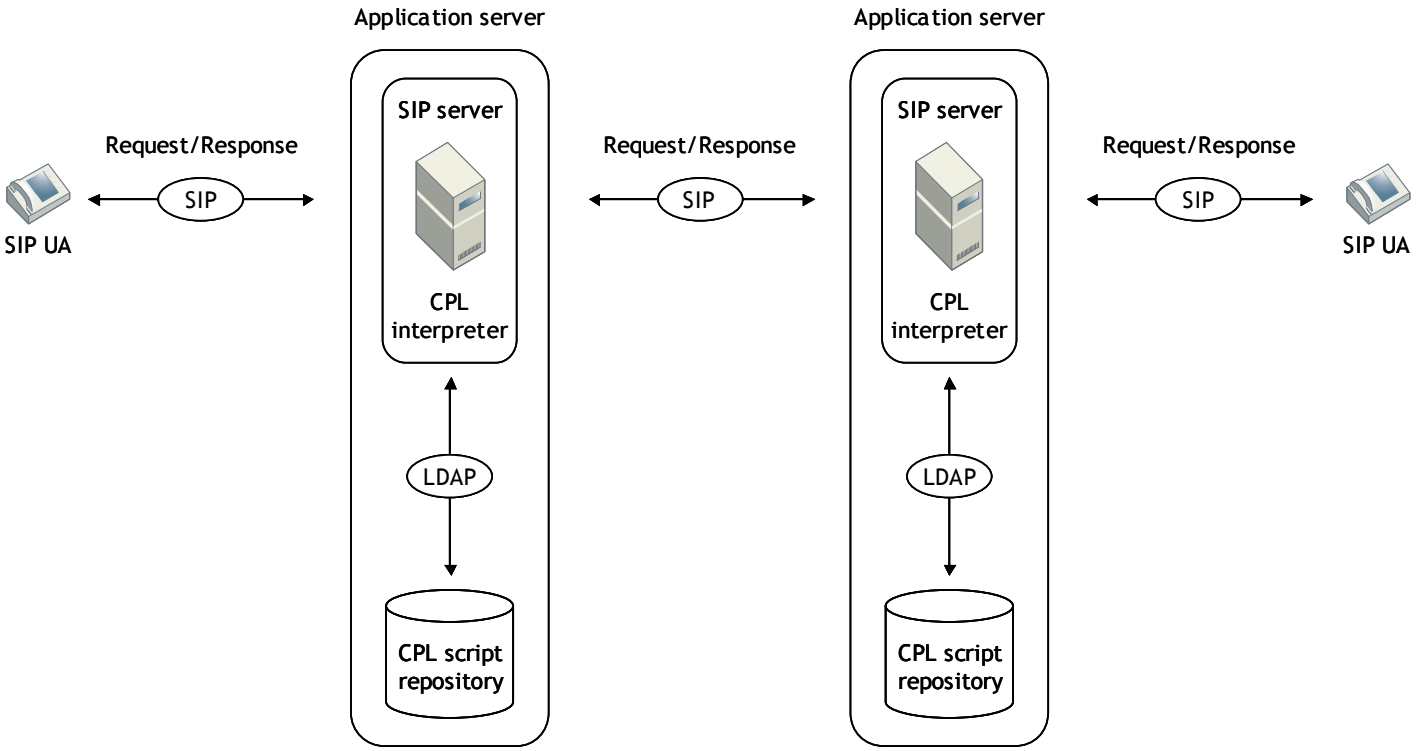


Call Processing Language (CPL)

- A language that describes and controls an Internet telephony service using SIP (or H.323)
- Defined by IETF IP Telephony (IPTEL) working group in RFC 3880
- Originally meant also for untrusted end-users and therefore the language is quite restricted
  - No assignment of variables
  - No loops or recursions
  - No ability to run external programs
- A CPL script is run in a SIP server, and it controls that system's proxy, redirect, and rejection actions for the set-up of a call

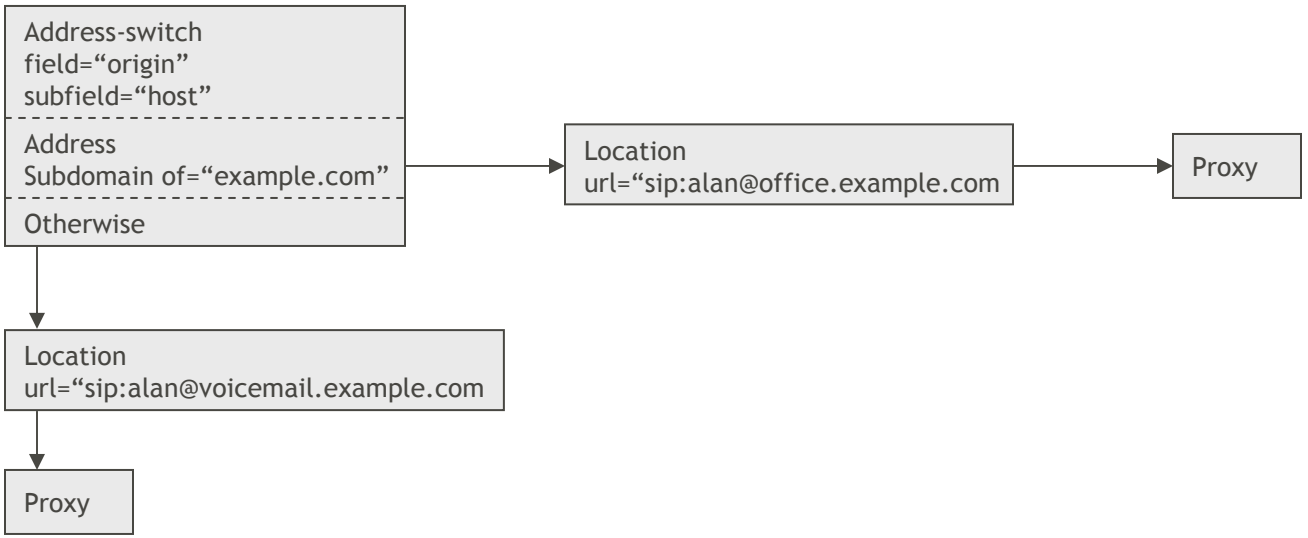
CPL cont.

- During the set-up of a call, scripts can be executed multiple times in different SIP servers (e.g. originating and terminating domains)



CPL cont.

- Based on XML
- Best modelled with a decision tree, which consists of nodes and outputs
- Nodes represent the actions that the script is able to make
- Outputs represent the result of corresponding action



CPL cont.

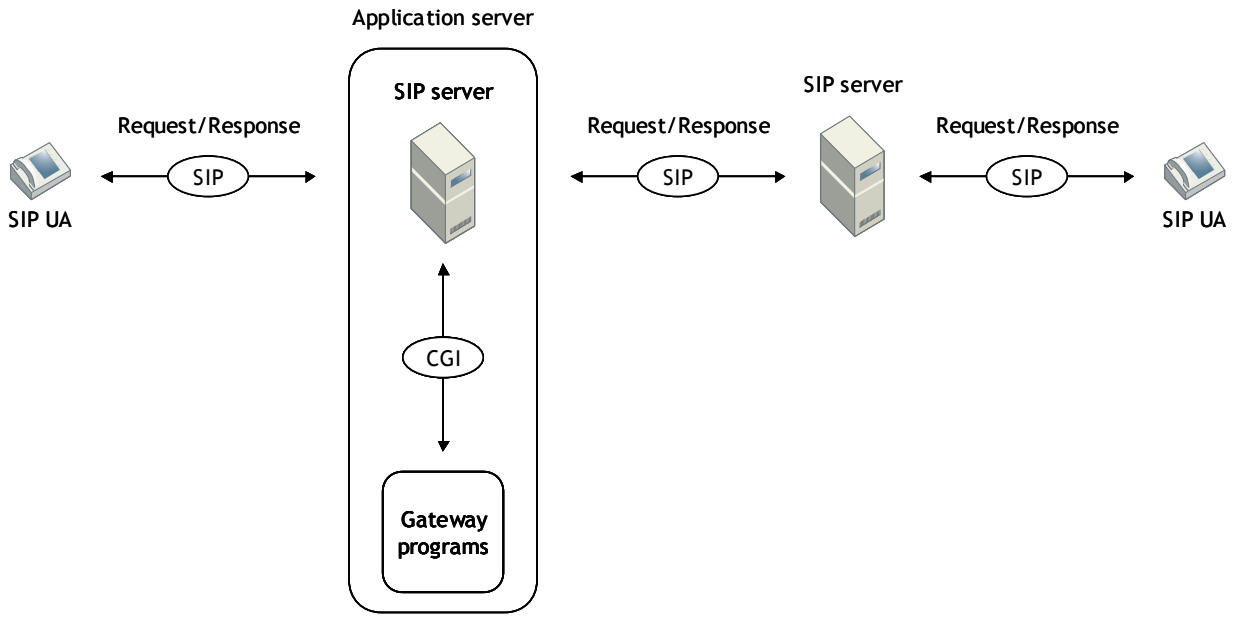
```
<cpl>
  <incoming>
    <address-switch field="origin" subfield="host">
      <address-subdomain-of="example.com">
        <location url="sip:alan@office.example.com">
          <proxy/>
        </location>
      </address>
      <otherwise>
        <location url="sip:alan@voicemail.example.com>
          <proxy/>
        </location>
      </otherwise>
    </address-switch>
  </incoming>
</cpl>
```

Common Gateway Interface for SIP (SIP-CGI)

- CGI is one tool that allows a web developer to create dynamic content on a web site
- CGI is a standard interface between external applications and web servers
- Due to the similarities between SIP and HTTP, CGI seems to be an attractive way to create services in a SIP enabled network
  - A lot of capable developers available
  - Components used in HTTP-CGI can be reused with SIP-CGI
- Other characteristics of SIP-CGI
  - Exposes all headers of a SIP request
  - All parts of requests and responses can be modified
  - Ease of extensibility
  - Virtually any programming language can be used

SIP-CGI cont.

- A SIP-CGI script is run in a SIP server
- A single script that is always executed, or multiple scripts, from which the target script is chosen by some parts of the header



SIP-CGI cont.

- The script gets input from environment variables and standard input
- To return data back to the SIP server process the script writes data to its standard output
  - The output consists of action lines and any number of optional SIP-CGI and SIP header fields

```
CGI-PROXY-REQUEST sip:alan@example.com SIP 2.0
Contact:
Subject: Earth's rotation

SIP/2.0 180 Ringing
```

Service triggering and charging

- Service triggering and charging are both an integral part of an IN service
- Neither CPL nor SIP-CGI specifications thoroughly speak out, how script triggering is done – it is up to the local policy at the application server
- SIP specifications do not address how charging should be carried out in SIP enabled networks and even today, there is no consensus in the industry, how VoIP calls should be charged
  - Therefore charging issues were left out of the scope of this study

Applied IN functions

- 10 existing IN functions were chosen in order to see, if CPL or SIP-CGI could be used to implement these functionalities in IP network

IN FUNCTION	CPL implementation	CGI implementation
A-number translation	NOK	OK
B-number translation	NOK	OK
Call barring	OK	OK
Short numbers	NOK	OK
Call forwarding	OK	OK
Allocation	NOK	OK
Time of day routing	OK	OK
Announcement playing	NOK	OK
Forking	OK	OK
Call gapping	NOK	NOK

Conclusions

- CPL is far too concise and simple to replace IN
  - No state maintained across a session
  - SIP messages cannot be directly modified
- SIP-CGI is a very versatile service creation technique
  - Uses general purpose programming language
  - Exposes all header information
  - The interface itself is quite cumbersome, because standard input/output and environment variables are used
- CPL and SIP-CGI are complementary to each other
  - CPL suitable for simple services such as call forwarding (also for end-users)
  - More demanding services are created with SIP-CGI

TeliaSonera