



Internet Telephony: **not** Telephony over Internet

Jonathan Rosenberg

Bell Laboratories

Spring VoN 99



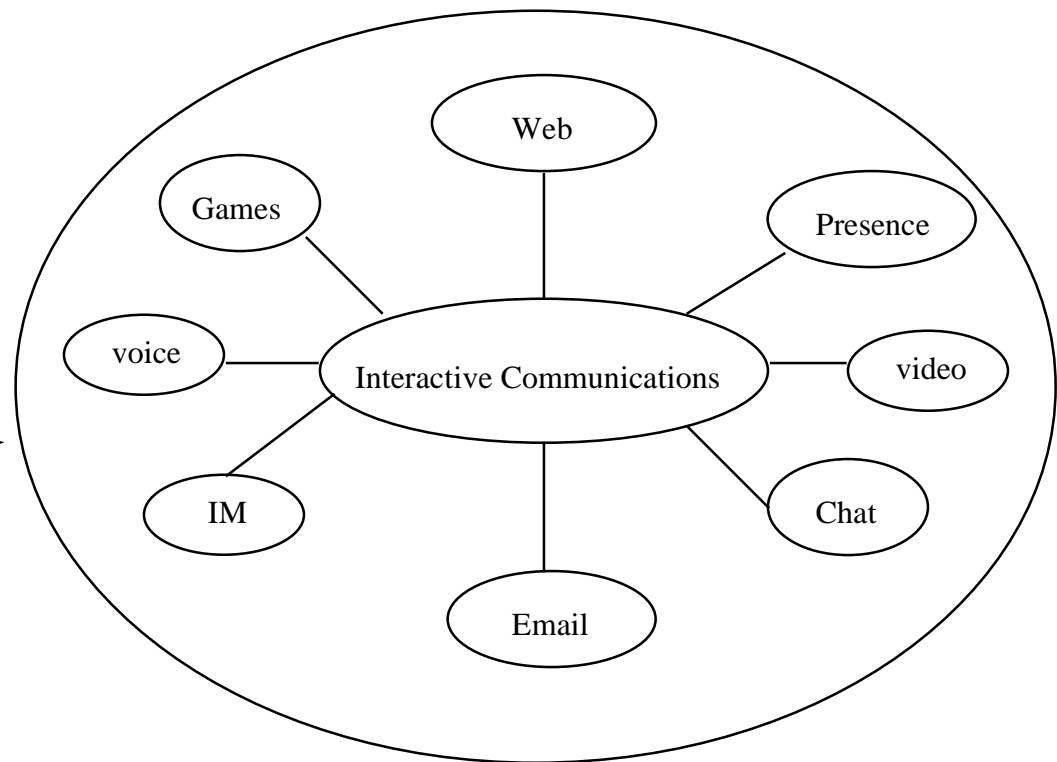
Telephony over Internet

- Emulation of Telephony Services on Internet
 - dumb end terminals (cable modems)
 - 12-digit keypad UI
 - transparency of services
 - it is important!
- Primary motivation
 - cost savings
 - non-telcos can enter
- Cost savings are transient
- Whats in it for *customers????*



Internet Telephony

- Integrate telephony services with
 - web
 - email
 - instant messaging and presence
 - text chat
 - interactive games



INTERNET TELEPHONY



New Services

- Integration causes *service multiplication*
 - 20 voice services X 20 web services = 400 integrated service possibilities
 - not all make sense
 - New services = revenue opportunities
- Examples
 - IM Notify when busy
 - subscriber gets instant messages when friends telephones (IP or POTS) available
 - Call redirect to web
 - web page returned instead of busy signal
 - Web IVR
 - web page of menus, final choice rings phone



More Services

- Shared web browsing
 - talk and browse jointly
- Transfer to email
 - Caller is disconnected and mail tool pops up
- Email call logs
 - Unanswered calls cause email notifications
- IM notifications of conference join
 - On a conference bridge, instant message indicates participant joins/leaves
- Web call-ID
 - web page of caller pops up when phone rings



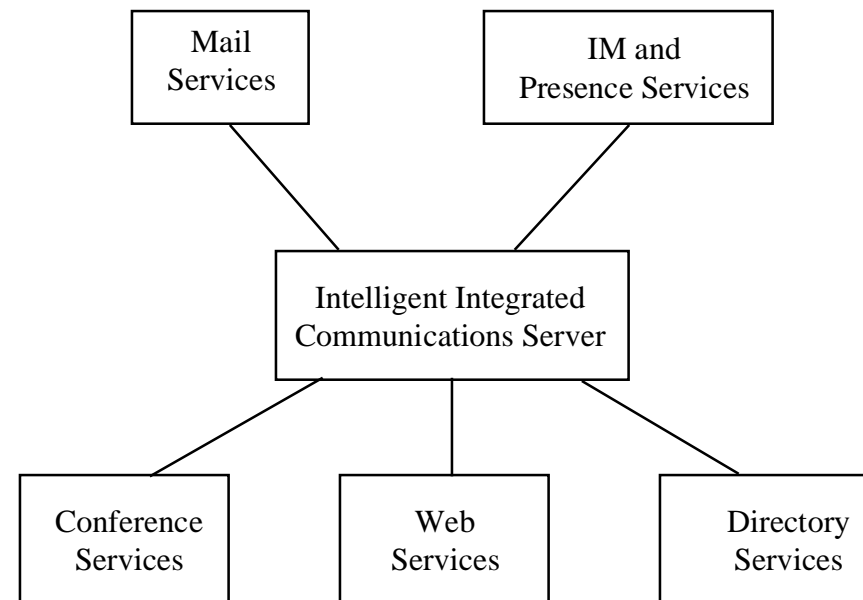
Who can get services?

- Advanced services can be offered to PSTN end systems too!
- VXML consortium
 - technology for providing web content on phone
 - allows web services to be exposed
- Speech to text
 - email sending
 - web browsing
 - IM
- Text to speech
 - Instant messages



How to do it?

- Integrated services = integrated server
 - SIP server/gatekeeper
 - SMTP/IMAP/POP client and server
 - Presence and IM server
 - Web access
 - Conference services access





Challenge: Programmability

- Explosion of features
= difficulty to control
- Need powerful,
flexible interfaces
 - reuse existing
interfaces
 - separate logic from
execution
 - engineer for flexibility
- Who can program
services?
 - Service providers
 - Third parties
 - End users!
- Approaches
 - SIP CGI
 - CPL
 - CORBA/DCOM?



SIP CGI

- SIP and HTTP are close cousins
- SIP CGI nearly same
 - Allow for “persistent” scripts
 - Allow script to proxy and return responses
- Benefits
 - ANY programming language
 - Loose coupling with server
 - separate process
 - IPC by file descriptors
 - no rebuilds or out of service
 - Little SIP understanding needed



Call Processing Language

- A scripting language to describe call services
- Language properties guarantee safety
- Based on XML
- Hand or tool authoring
- Example: “call forward on busy/no-answer”
- Under development in IETF

```
<call>
  <location url="sip:jones@pc.ex.com">
    <proxy timeout="8s">
      <busy>
        <location url="sip:jones@vmail.ex.com"
          merge="clear" id="voicemail" >
          <proxy />
        </location>
      </busy>
      <noanswer>
        <link ref="voicemail" />
      </noanswer>
    </proxy>
  </location>
</call>
```



Conclusion

- Integrated services is key
- Require a scalable, integrated server
 - web, email, SIP, messaging services
- Requires a powerful way to program it
 - CGI, CPL