telefaks* application server for FreeSWITCH

Peter Steinbach
Mein50Plus GmbH
Theo-Geisel-Str. 25
Usingen, Germany, 61250
Tel.: +49 6081 688 533
www.telefaks.de
Information@telefaks.de
Introduction

- This presentation is all about telephony services for
  - VoIP
  - POTS
  - Google Talk / Jabber
  - Messaging

- Building telephony services based on
  - OpenSource software
  - Standard server hardware
Who we are

• Coming from Asterisk
• On Freeswitch since beg. of June 2008
• Transferred all our applications to Freeswitch since then
• Strong focus on
  – Integrating Freeswitch
  – Ruby and Rails Development
  – Encryption
Freeswitch- the new swiss knife for VoIP (1)

- FreeSWITCH is a new alternative to Asterisk
- Developed by people who wanted to have a better code base compared to Asterisk and a better and more flexible structure

**Advantages**
- Call volume per server (3000+)
- Configuration by XML instead of sometimes difficult Asterisk-Syntax
- Higher stability at high call volumes
- Better central administration by webservices
- Several virtual PBXs on one server
- Simpler call routing in bigger installations
- Encryption via TLS and SRTP (currently the only OpenSource solution)

**Disadvantages**
- General available GUI missing, configuration via XML files
- Not as established on the market compared to Asterisk (but more stable in produktion)

**Outlook**
- Will become one of the standards for larger installations
Freeswitch- the new swiss knife for VoIP (2)

Can be used as:

• VoIP-Switch
• VoIP-Router
• IVR-System
• Phone conference server
• PBX
• B2BUA(Back to back user agent)
• Session border controller
• Basic Topology Hiding Session Border Controller,
• Application Server (VoiceMail, Konferenz, IVR)
• Integration platform
• Register proxy
Freeswitch- the new swiss knife for VoIP (3)

Availability:
- Mostly all Linux platforms
- Sun Solaris / OpenSolaris
- Windows
- Mac OS X
- BSD
Freeswitch - the new swiss knife for VoIP (4)

**Key points**

- Scalability
- Built-in redundancy mechanisms
- Supports a number of communication protocols (incl. Jabber und Skype)
- Encryption of Voice (SRTP) and call setup (TLS)
- Voice codecs up to 48KHz
- A number of interfaces for configuration and call control (synchronous and asynchronous), perfect for dynamic call routing
- Word recognition (Sphinx)
- Text-To-Speech via Cepstral TTS
Freeswitch-Highlights (1)

Skalability

- ~ 3000 simultaneous Calls including media
- Factor >> 10 with media outside Freeswitch
- Built-in redundancy mechanisms via XML-Curl for configuration and call control
telefaks* application server
Why an application server framework?

- Our Freeswitch projects usually have a larger scale than e.g. an Asterisk PBX
- A single Freeswitch is per default configured by XML files
- On top there exists a number of interfaces for configuration and synchronous/asynchronous call control
- Integrating large projects therefore requires a lot of groundwork to be done
- Some nice GUIs exist already, each one targeting a dedicated scenario (e.g. PBX, Callcenter)
- however, a system which will cover all scenarios by 100% will most probably never exist
We need a framework to abstract functionalities for integrating large Freeswitch projects.
What is basically needed for that?

- Administration GUI
- Handling of more than one freeswitch server
- Customer hierarchies
- IVR functionalities
- Callcenter support
- Asynchronous call handling
- Realtime interface with web browser (e.g. push status)
What is it built of

- Freeswitch of course
- some Ruby processes for interfacing with Freeswitch
- Ruby on Rails for the web interface
- Javascript and AJAX for the web interface
- a bit of LUA
- a push server
What ist covers

- Support of multiple Freeswitch servers
- Basic PBX functionalities (is needed almost everywhere)
- Conferencing (setup and „live“ management)
- Call Queues
- Callback/dialthru
- IVR State machine with setup via GUI
- Callcenter workflows with direct interaction between browser and freeswitch
- TTS and ASR Support
- Encryption of calls (TLS/SRTP)
- Complex routing algorithms for larger networks
- Prepared for billing functionalities
- Channel Spy
- Custom applications
- Interface to SyncML
How it's designed

User-Interface (Webbrowser)

HTTPS
Call Routing
Event Handling
Asynchronous Handling
State machine

Base Logic

Application-Server

SyncML
RSS
Other Services
Other databases
Wikipedia

XML-Curl
Event-socket
Event-socket
Event-socket
PBX functionalities
Sample PBX functionalities

- Serve multiple clients
- Clients can be spread over multiple instances of Freeswitch
- User administration with client hierarchies
- Management of SIP endpoints
- Voicemail
- Call forwarding (parallel + sequential hunting)
- Short numbers for each endpoint
- One-time numbers (or n times usage), obfuscated numbers
- Dialthru/Callback
- Special numbers
- Conferences
- Call queues
- Encryption TLS/SRTP
- ... more
Sample PBX functionalities

Telefaks Freeswitch Management

Editing directory

Customer id
Telefaks

Exten
83533

Password

Gateway
tips.telefaks.biz

Full name
Peter Steinbach

External CTD
0608688533

Enable direct callforward
F

Direct callforward to
0608688533

Voicemail
83533

Vm Password

Vm-Email
steinbach@telefaks.biz

SIP registrations

Available Numbers

for exten number

<table>
<thead>
<tr>
<th>Customer</th>
<th>Exten_from</th>
<th>Exten_to</th>
</tr>
</thead>
<tbody>
<tr>
<td>Telefaks</td>
<td>83533</td>
<td>83533999999</td>
</tr>
<tr>
<td>Internal</td>
<td>99999</td>
<td>99999</td>
</tr>
<tr>
<td>MeinE6Plus</td>
<td>26824</td>
<td>26824999999</td>
</tr>
<tr>
<td></td>
<td>83533</td>
<td>83534999999</td>
</tr>
</tbody>
</table>
Sample Conferencing functionalities

- Conference definition

### Telefaks Freeswitch Management

#### Editing conference

**Host**
- sip5.telefaks.biz

**Customer**
- Men50plus

**Conference description**
- Sales

**Conference type**
- Conference 8kHz en ComfortNoise EnergyLevel 3000

**Valid from**
- September 2008 - 13

**Valid to**
- November 2009 - 13

**Active**
- Yes

**Pin**
- 7777777777777777777777777

**Kick all members out of the conference after initiator hangs up**
- No

**Record whole conference**
- No

#### Conference Numbers to invite

<table>
<thead>
<tr>
<th>No</th>
<th>Extension</th>
<th>Active</th>
<th>Originator</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>035331</td>
<td></td>
<td></td>
</tr>
<tr>
<td>2</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>06081688533</td>
<td></td>
<td></td>
</tr>
<tr>
<td>4</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>5</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>6</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>7</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

#### Available number ranges

<table>
<thead>
<tr>
<th>Customer</th>
<th>Range_from</th>
<th>Range_to</th>
</tr>
</thead>
<tbody>
<tr>
<td>Telefaks</td>
<td>83533</td>
<td>83533999999</td>
</tr>
<tr>
<td>Internal</td>
<td>99998</td>
<td>99998</td>
</tr>
<tr>
<td>Men50plus</td>
<td>26824</td>
<td>26824999999</td>
</tr>
</tbody>
</table>

*Range_from* is the inclusive range of numbers that can be dialed to join the conference. *Range_to* is the inclusive range of numbers that can be dialed to make calls from within the conference.
Sample Conferencing functionalities

- Conference live management

Telefaks Freeswitch Management

Listing active_conferences

<table>
<thead>
<tr>
<th>Conference Name</th>
<th>Record conference</th>
<th>Conference lock</th>
<th>Conference PIN</th>
<th>Send data to all members</th>
<th>Invite into conference</th>
</tr>
</thead>
<tbody>
<tr>
<td>89533200</td>
<td></td>
<td></td>
<td>PIN</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Conference members

<table>
<thead>
<tr>
<th>Member</th>
<th>Member’s Speaker</th>
<th>Member’s Microphone</th>
<th>Energy Level</th>
<th>Kick out</th>
<th>Send data to this member</th>
<th>Transfer member</th>
</tr>
</thead>
<tbody>
<tr>
<td>TLS</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>SRTP</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>895333</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>TLS</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>SRTP</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Sample PBX functionalities

- Operator Panel (still in development, Jan / 2010)
  - similar to „Flash Operator Panel“ for Asterisk
  - initiate, answer, transfer and drop calls via „Drag and Drop“

(see example videos)
IVR functionalities
IVR Callback and Callthru application

Step 1: Draw the workflow

Goal:
- Identify client/caller
- Hangup, then store callback number if client is callback customer
- Next step: callback to the client
- Offer to enter target number via DTMF and connect the call
IVR Functionalities

- Built-in state machine for defining IVRs and other workflows
- IVRs are defined the following way:
  - Step 1: Draw the callflow as UML state diagramm
    - define actions
    - define transitions
  - Step 2: Upload UML state diagram to the application server
  - Step 3: Specify actions for each state on the web GUI
  - Step 4: Test the state machine on the web GUI (html)
  - Step 5: Take the state machine into production (now with voice)
IVR Functionalities

• Interaction with the caller
  – Play sound files or external sound streams (play multiple files and variables)
  – Text to speech
  – Read DTMF
  – Voice menus (DTMF)
  – Record users voice and playback later
  – Word recognition (ASR)
  – Answer a call
  – Hangup a call
  – Dial a number
  – Transfer a call
  – Numerous customized actions

• ++ Numerous asynchronouse actions during a call

• early media mode for some actions
### IVR Callback and Callthru application

**Step 2: Specify actions in detail**

<table>
<thead>
<tr>
<th>Action name</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>ask_destination_no</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Tts text</th>
</tr>
</thead>
<tbody>
<tr>
<td>Please enter the destination Number (10 or 11 digits)</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Voice files (separate multiple sound files by linefeeds)</th>
</tr>
</thead>
<tbody>
<tr>
<td>welcome.wav</td>
</tr>
<tr>
<td>you-are-using.wav</td>
</tr>
<tr>
<td>$services.wav</td>
</tr>
<tr>
<td>please-enter-num-to-call.PCMU</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Do TTS? (Otherwise play sound files)</th>
</tr>
</thead>
<tbody>
<tr>
<td>✗</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>No of Digits when asked for Input</th>
</tr>
</thead>
<tbody>
<tr>
<td>11</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Interruptable bykeypress?</th>
</tr>
</thead>
<tbody>
<tr>
<td>✗</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Hear Params</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
</tr>
</tbody>
</table>
IVR Callback and Callthru applikation
Step 2: Test workflow on the web browser

State: 128312 "Ask destination No"

Compare: No conditions

Executed: ask_destination_no|destination

Input:

Play zigit/welcome.wav
Play zigit/you-are-using.wav
Play zigit/callback.wav
Play zigit/please-enter-num-to-call.PCMU

Input: [ ] submit
Callcenter functionalities
Callcenter application framework

- Extension to IVR Application
- Webbrowser initiates actions on Freeswitch
- Freeswitch pushes data to the web browser (AJAX push services)

Interactions to Freeswitch
- Dial a number from a database
- Answer a call
- Play messages
- Start recording
- Stop recording
- Forward call
- Hangup Call

- Push services to the web browser
- Show status of a call
- Alert incoming calls
- Open CRM window
Sample callcenter application:
Step 1: Define Workflow

Get new number from the database

User input defines next steps

Save to database via database profiles
Sample callcenter application:  
Step 2: Define Forms

**Callcenter form assistant**

Enter variable name to be used in processing:
- favourite

Please select control type:
- select

Enter description text for this control:
- What is your favourite brand to buy a vehicle?

Enter parameters for this control:
- Volvo, Mercedes, DAF, Iveco, MAN, Toyota, VW, Audi, BMW

Enter default options for this control:
- Volvo

**Form preview**

Ask for new procurements

When are you planning to buy your next vehicles?
- not_planned_yet
- 2009
- 2010
- 2012
- 2013
- later

What is your favourite brand to buy a vehicle?
- Volvo

Customer history:

- Date
- User
- Campaign
- Follow-up
- Comment

Define new form elements

Preview new form
Sample callcenter application:
Step 3: Run workflow

Telefaks Freeswitch Management

Callcenter form state: 128626 "Ask next Data"

- Compare: No conditions
- Executed: form_step_5

Ask for new procurements

When are you planning to buy your next vehicles?
- not_planned_yet
- 2009
- 2010
- 2012
- 2013
- later

What is your favourite brand to buy a vehicle

- Volvo

Customer history:

<table>
<thead>
<tr>
<th>Date</th>
<th>User</th>
<th>Campaign</th>
<th>FollowUp</th>
<th>Comment</th>
</tr>
</thead>
<tbody>
<tr>
<td>2009-05-09</td>
<td>peter</td>
<td>poll_cars</td>
<td></td>
<td>Completed</td>
</tr>
<tr>
<td>2009-05-08</td>
<td>peter</td>
<td>poll_cars</td>
<td>2009-06-09</td>
<td></td>
</tr>
<tr>
<td>2009-06-07</td>
<td>peter</td>
<td>poll_cars</td>
<td>2009-06-08</td>
<td></td>
</tr>
</tbody>
</table>
Push services
Push services

- every GUI user has an assigned phone number
- web browser registers on this phone number
- web browser gets status pushed from Freeswitch
  - Example: successful hangup

<table>
<thead>
<tr>
<th>Status</th>
<th>Incoming Call from: 723321 (Peter Steinbach FS) and IP 217 11.186</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Event: CHANNEL_EXECUTE_COMPLETE, state: CS_HANGUP</td>
</tr>
</tbody>
</table>

Telefaks Freeswitch Management

### Listing directories

#### Phone Numbers/Conferences

- **Phone numbers**
  - New directory entry (exten)

#### Incoming call:

<table>
<thead>
<tr>
<th>Status</th>
<th>Call from: 835333 and IP 217 186</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Event: CHANNEL.PROGRESS, state: CS_CONSUME_MEDIA</td>
</tr>
</tbody>
</table>

#### Active call:

<table>
<thead>
<tr>
<th>Status</th>
<th>Call from: 723321 (Peter Steinbach FS) and IP 217 186</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Event: CHANNEL_ANSWER, state: CS_EXECUTE</td>
</tr>
</tbody>
</table>
Customizing your application
Call Routing with regular expressions

Editing routing

Profile
1

Description
German Provider QSC International

Active
No

Dialplan
Next Gateway insecure

Context from
default

Gateway from
sip5.telefaks.biz

Time Interval
whole day

Sort Id
0

Number From (enter regular expression)
^\(00\)[1-9]\(\{4,13\}\)$

Context to
default

Gateway to
QSC-07141

Number To (enter fixed number, or $1 for the dialled number or $target_number$ for the replaced number)
$target_number$

Number To Match (Regex which shall apply to the finally dialled number, leave empty if no change shall apply)
^\(00\)[1-9]\(\{4,13\}\)$

Comment
Strip off 00 from 0049xxx
Call handling via templates

<!-- start a generic conference with the settings of the "default" conference profile -->
<!-- Target No $target_number$ -->
<extension name="conference $conference_name$">
  <condition field="destination_number" expression="^\(\d+\)"/>
  <action application="set" data="dialplan_comment=$dialplan_comment$"/>
  <!-- this is filled up with external participants and a hangup hook if needed -->
  $conference_inivitations$
  <action application="answer"/>
  <action application="send_display" data="Conference $1"/>
  <action application="conference" data="$conference_number@$context$"/>
</condition>
</extension>

• Application server defines additional variables
• Variables are expanded at runtime
Customizing your own applications


- Special numbers can be used to trigger own dialplan actions
- dialplan actions can be XML templates or customized Ruby code

**Telefaks Freeswitch Management**

**Editing special_number**

- **Customer**
  - Telefaks
- **Number**
  - 83533405
- **Description**
  - Speak Wikipedia
- **Active**
  - ✔
- **Show on panel**
  - ✔
- **Dialplan**
  - Execute (select Execute dialplan before when using)
  - Custom.speak_wikipedia("Frankfurt am Main")

**Available number ranges**

<table>
<thead>
<tr>
<th>Customer</th>
<th>Range from</th>
<th>Range to</th>
</tr>
</thead>
<tbody>
<tr>
<td>Telefaks</td>
<td>83533</td>
<td>8353399999</td>
</tr>
<tr>
<td>Internal</td>
<td>99998</td>
<td>99998</td>
</tr>
<tr>
<td>Telefaks_public</td>
<td>99999999900</td>
<td>99999999999</td>
</tr>
<tr>
<td>Main50Plus</td>
<td>25824</td>
<td>2582499999</td>
</tr>
<tr>
<td>Telefaks_private</td>
<td>83534</td>
<td>8353499999</td>
</tr>
</tbody>
</table>
def self.speak_wikipedia(search_exp):
    text=self.get_wikipedia_text(search_exp)
    master="<action application="speak" data="cepstral|katrin|$text$"/>
    erg="<!-- Wikipedia entry to speak: '#{search_exp}' -->\n    if text
        text.each do |line|
            if !line.strip.empty?
                erg+=master.gsub("$text$", line)
            end
        end
    end
    erg
</!
"
Some examples for customizing

- Wikipedia as shown before
- Speak selected content of news sites
- Speak RSS feeds
- Speak file contents
- Speak meter values from external interfaces
- Access calendar from SyncML (Funambol)
- Intercom, global announcements
- Reverse internet CID lookup
Performance

• using caching techniques wherever applicable
  – „Memcache“ allows distributed caching over multiple servers
• Tested under High Load
  – up to 250 call setups per second out of the box on a Dual Core AMD 2,5GHz (caching enabled)
  – up to 160 call setups per second out of the box on a Dual Core AMD 2,5GHz (caching disabled)
• Outlook:
  – scales well with the number of processors (processes are CPU intensive)
  – scales well with the number of machines (http cluster techniques used)
  – Further performance improvement with Ruby 1.9 and optimized, self-compiled Ruby binaries
Thank you!

Peter Steinbach
steinbach@telefaks.biz

Hans-Jürgen Bornhorst
bornhorst@telefaks.biz