# Table of Contents

- What is SIP Communicator?
- SIP Communicator Overview – Currently supported and planned features.
- Deployment, customization and maintenance – Extensibility, ease of maintenance and deployment.
- Creating plug-ins for SIP Communicator
SIP Communicator Overview

SIP Communicator is an open source (LGPL) Audio/Video software phone and instant messenger. Among others, we currently support:

- **Audio and Video calls with SIP**
- Instant messaging with Jabber, ICQ/AIM, Yahoo! Msngr and MSN
- IPv6 support for SIP and Jabber
- Support for multiple accounts and meta contacts
- Basic NAT & Firewall Traversal with STUN.
- Modularity, extensibility, and flexibility with OSGi
- Platform specific installers for Windows, Debian, Fedora, Mac OS X

Visit [http://sip-communicator.org](http://sip-communicator.org) for more details.
SIP Communicator Overview
SIP Communicator Overview
Instant Messaging
SIP Communicator Overview
Instant Messaging
SIP Communicator Overview

Instant Messaging History

[FULL IMAGE OF HISTORY WINDOW]

FOSDEM, February 25th 2007
Features Currently in Progress

- **Powerful and scalable firewall support** – Combine technologies like TURN, STUN, ICE and P2P in order to achieve powerful, secure, and scalable firewall support.

- **Robust and Seamless IPv6 support** – We aim to achieve more than a mere “support for IPv6”. We would like to guarantee its transparent and seamless usage. In other words, usage of IPv6 should not require any special configuration and should not cause any problems.

- **Security** – Encrypt all media. Usage of P2P for firewall traversal imposes reliable security of all media transmitted by the SIP Communicator in order to prevent relaying nodes from eavesdropping.

- **Automatic updates and one-click plug-in installation** - Provide a user interface and an online repository for SIP Communicator plug-ins.

- **Other plug-ins** - IRC, shared whiteboards, multiparty video and chat conferencing.
A Look Inside – The OSGi Core

OSGi Framework

- Jabber
- SIP
- MSN
- Media
- User Interface
- Message History
- Meta Contact List
- Firewall Traversal
- ICQ
SIP Communicator is built upon the Apache Felix implementation of the OSGi framework. This helps us provide qualities such as:

- **Modularity & Flexibility** – All components of SIP Communicator are implemented as separate, replaceable modules. It is possible to run the application with different sets of features and functionalities depending on the intended user or target platform.

- **Extensibility** – It is very easy to implement additional features in the form of plug-ins. Developers that are new to SIP Communicator could easily start developing for it since they only need to get acquainted with existing APIs and not the entire source code.

- **Ease of maintenance and deployment** – The concept of an OSGi Bundle Repository allows SIP Communicator users to download, install and configure new plug-ins with a few clicks.
A Look Inside – Modularity & Flexibility

SIP Communicator is built upon the Apache Felix implementation of the OSGi framework. This helps us provide qualities such as:

- **Modularity & Flexibility** – All components of SIP Communicator are implemented as separate, replaceable modules. It is possible to run the application with different sets of features and functionalities depending on the intended user or target platform.

- **Extensibility** – It is very easy to implement additional features in the form of plug-ins. Developers that are new to SIP Communicator could easily start developing for it since they only need to get acquainted with existing APIs and not the entire source code.

- **Ease of maintenance and deployment** – The concept of an OSGi Bundle Repository allows SIP Communicator users to download, install and configure new plug-ins with a few clicks.

FOSDEM, February 25th 2007
SIP Communicator is built upon the Apache Felix implementation of the OSGi framework. This helps us provide qualities such as:

- **Modularity & Flexibility** – All components of SIP Communicator are implemented as separate, replaceable modules. It is possible to run the application with different sets of features and functionalities depending on the intended user or target platform.

- **Extensibility** – It is very easy to implement additional features in the form of plug-ins. Developers that are new to SIP Communicator could easily start developing for it since they only need to get acquainted with existing APIs and not the entire source code.

- **Ease of maintenance and deployment** – The concept of an OSGi Bundle Repository allows SIP Communicator users to download, install and configure new plug-ins with a few clicks.
SIP Communicator is built upon the Apache Felix implementation of the OSGi framework. This helps us provide qualities such as:

- **Modularity & Flexibility** – All components of SIP Communicator are implemented as separate, replaceable modules. It is possible to run the application with different sets of features and functionalities depending on the intended user or target platform.

- **Extensibility** – It is very easy to implement additional features in the form of plug-ins. Developers that are new to SIP Communicator could easily start developing for it since they only need to get acquainted with existing APIs and not the entire source code.

- **Ease of maintenance and deployment** – The concept of an OSGi Bundle Repository allows SIP Communicator users to download, install and configure new plug-ins with a few clicks.
SIP Communicator is built upon the Apache Felix implementation of the OSGi framework. This helps us provide qualities such as:

- **Ease of maintenance and deployment** – The concept of an OSGi Bundle Repository allows SIP Communicator users to download, install and configure new plug-ins with a few clicks.
SIP Communicator is built upon the Apache Felix implementation of the OSGi framework. This helps us provide qualities such as:

- Ease of maintenance and deployment – The concept of an OSGi Bundle Repository allows SIP Communicator users to download, install and configure new plug-ins with a few clicks.

FOSDEM, February 25th 2007
SIP Communicator is built upon the Apache Felix implementation of the OSGi framework. This helps us provide qualities such as:

- **Ease of maintenance and deployment** – The concept of an OSGi Bundle Repository allows SIP Communicator users to download, install and configure new plug-ins with a few clicks.
SIP Communicator 1.0 – Application Design

- **Protocol Service**
  - Jabber Impl

- **Abstract APIs**

- **Protocol Service**
  - SIP (jain-sip-ri)

- **Stats Plug-in**

- **History Impl**

- **Msg History**

- **Some Plug-in**

- **UI Service**
  - Java Swing

- **Media Service**
  - JMF

- **OSGi**
SIP Communicator 1.0 – Application Design

Abstract APIs

Protocol Service
  Jabber Impl

Stats Plug-in

Some Plug-in

OSGi

Protocol Service
  SIP (jain-sip-ri)

Media Service
  J MF

UI Service
  Java Swing

Bundle Specifics
SIP Communicator 1.0 Architecture

A Simple Scenario

Dude (Online)

UserInterfaceService

Swing UI Service mpl Bundle

network

FOSDEM, February 25th 2007
SIP Communicator 1.0 Architecture

A Simple Scenario

Dude (Online) _ o x
Hi dude, what r u up to?
Send

UserI nterfaceService
Swing UI Servicel mpl Bundle

network

FOSDEM, February 25th 2007
Hi dude, what r u up to?

Dude (Online)_ o x

1 `actionPerformed(event)`

UserInterfaceService

Swing UI Service Impl Bundle

OSGi Framework

network

ProtocolProviderService

Jabber Protocol Impl Bundle
SIP Communicator 1.0 Architecture

A Simple Scenario

1. `actionPerformed(event)`
2. `getService(providerRef)`
3. `sendMessage(Contact)`
4. `sendMessage()`

Dude (Online)

Hi dude, what u up to?

Send

UserInterfaceService

Swing UI Service Impl Bundle

ProtocolProviderService

Jabber Protocol Impl Bundle

Network

OSGi Framework

FOSDEM, February 25th 2007
SIP Communicator 1.0 Architecture
A Slighty More Complex Scenario (1)

FOSDEM, February 25th 2007
SIP Communicator 1.0 Architecture

A Slighty More Complex Scenario (1)

FOSDEM, February 25th 2007
A Slightly More Complex Scenario (2)

1. `getService(providerRef)`
2. `registerCallListener(this)`

SIP Communicator 1.0 Architecture

- OSGi Framework
- SIP Implementation Bundle
- Swing UI Service (Impl Bundle)
- UserInterfaceService
- ProtocolProviderService
- CallTransfer Plug-In

FOSDEM, February 25th 2007
A Slightly More Complex Scenario (3)

1 INVITE req received

2 incomingCallReceived(evt)

3 container.addComponent(cmp)

Alerting …
The Duke is calling you!

FOSDEM, February 25th 2007
SIP Communicator 1.0 Architecture

A Slightly More Complex Scenario (4)

Alerting …
The Duke is calling you!

+3592166166

Transfer

5 transferCall(call, dst)

4 actionPerformed(evt)

UserInterfaceService

Swing UI Service! mpl Bundle

ProtocolProviderService

SIP Implementation Bundle

network

6 send REFER req

FOSDEM, February 25th 2007
http://sip-communicator.org

SIP Communicator tutorials:

How to create OSGi Services for SIP Communicator:
http://www.sip-communicator.org/index.php/Documentation/CreatingServices

How to write GUI plugins:
http://www.sip-communicator.org/index.php/Documentation/HowToWriteGuiPlugins

How to implement support for your favorite protocol in SIP Communicator:
http://www.sip-communicator.org/index.php/Documentation/HowToImplementProtocols

How to create and maintain a SIP Communicator installer:
http://www.sip-communicator.org/index.php/Documentation/HowToBuildAnInstaller