

Asterisk Architecture

A glimpse under the hood

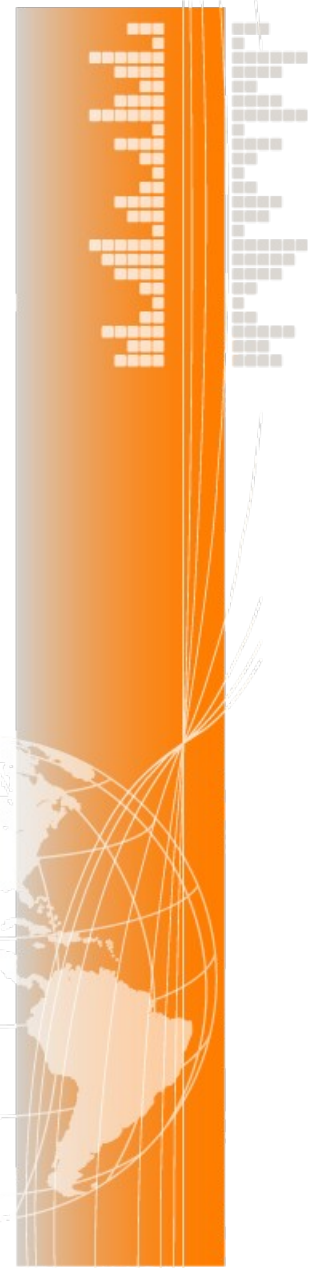
Joshua Colp
Software Developer
Digium, Inc.

What will be covered

- General Overview
- Incoming Call Flow
- Outgoing Call Flow
- Call Bridging
- Transfers (Attended/Blind)

General Overview

- Modularized – easy to plug new stuff in
- Abstracted – everything internally speaks the same



Module Types

- Channel Drivers
- Applications
- Dialplan Functions
- Resource Modules
- Codecs
- File Formats

Abstraction

- All module types have defined callbacks
- Each module must implement the defined callbacks
- Core API calls use the callbacks to interact

Channel Driver

- Protocol specific implementation
- Core interacts with it to talk to a device
- Each one implements the same callbacks

Channel

- No knowledge of underlying protocol
- Has link back to channel driver
- Contains audio/video format information
- Contains context/extension/priority information

PBX

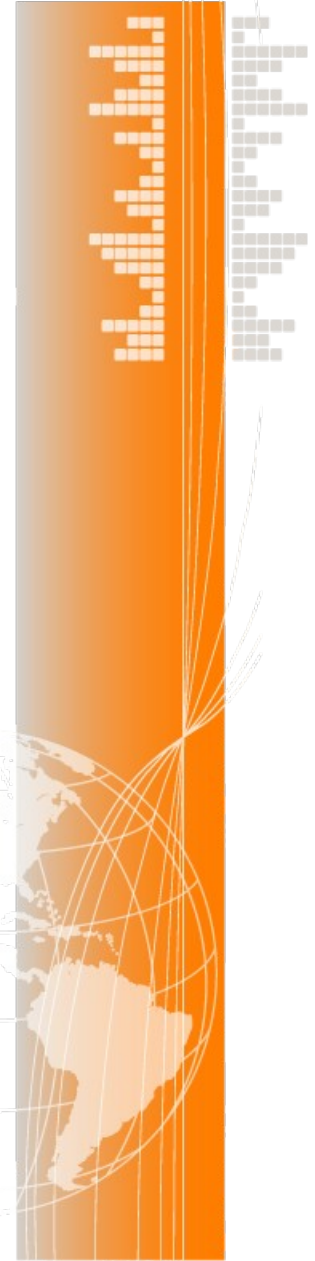
- Knows of contexts/extensions/priorities
- Executes applications/dialplan functions on channels
- Cleans up channel when it has hung up

Application

- No knowledge of underlying protocol
- Interacts with channel through standard API

Incoming Call (Simple)

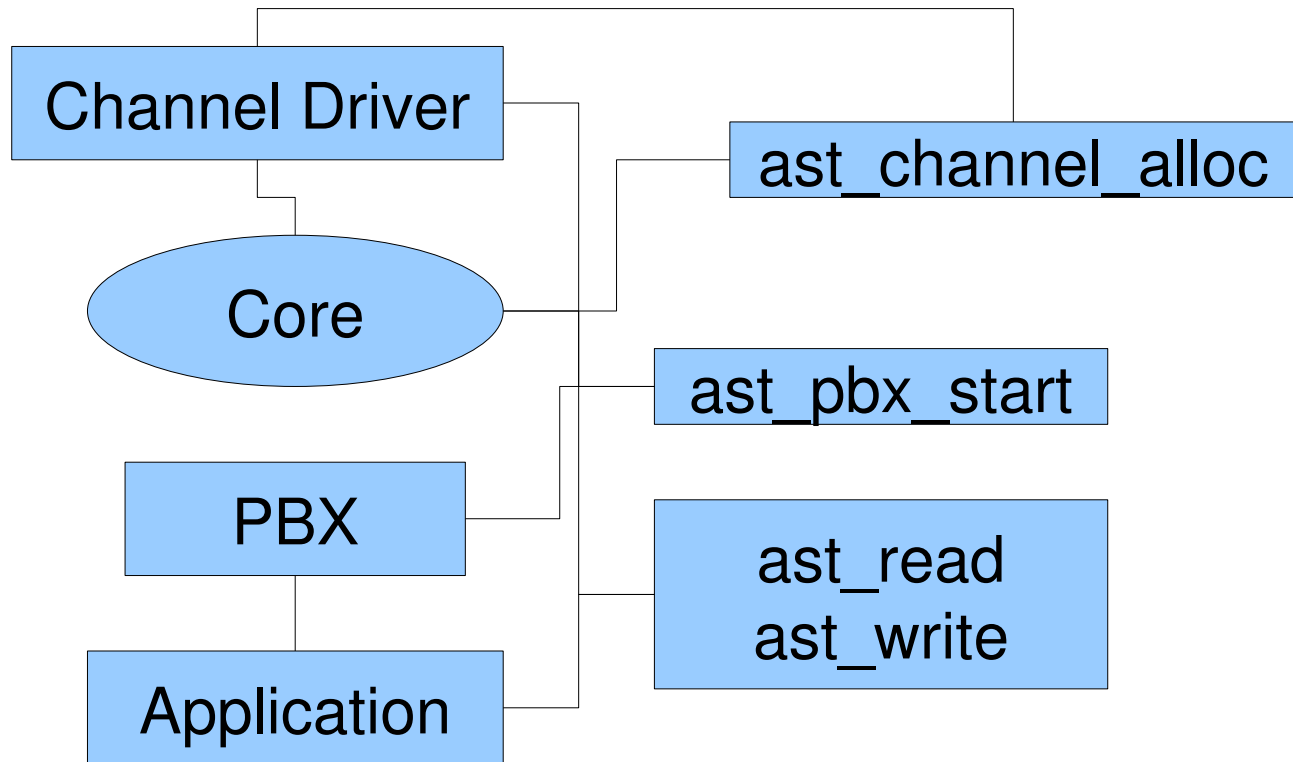
- Channel driver accepts call
- Asterisk channel is created
- Sent into dialplan
- Applications are executed on it
- Channel is hung up



Incoming Call (API)

- `ast_channel_alloc`
- `ast_pbx_start`
- `ast_read` calls `read`
- `ast_write` calls `write`
- `ast_hangup` calls `hangup`

Incoming Call (Diagram)



Outgoing Call (Simple)

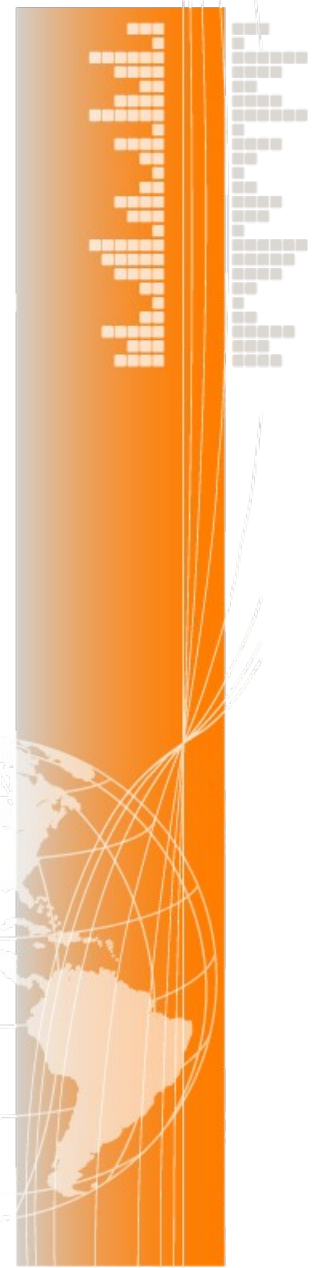
- Channel is requested from channel driver
- Channel is called
- Calling thread handles channel
- Channel is hung up

Outgoing Call (API)

- `ast_request` calls `requester`
- `ast_call` calls `call`
- `ast_read` calls `read`
- `ast_write` calls `write`
- `ast_hangup` calls `hangup`

Outgoing Call (Future)

- `ast_dial_new`
- `ast_dial_append`
- `ast_dial_run`
- `ast_dial_answered`
- `ast_dial_destroy`



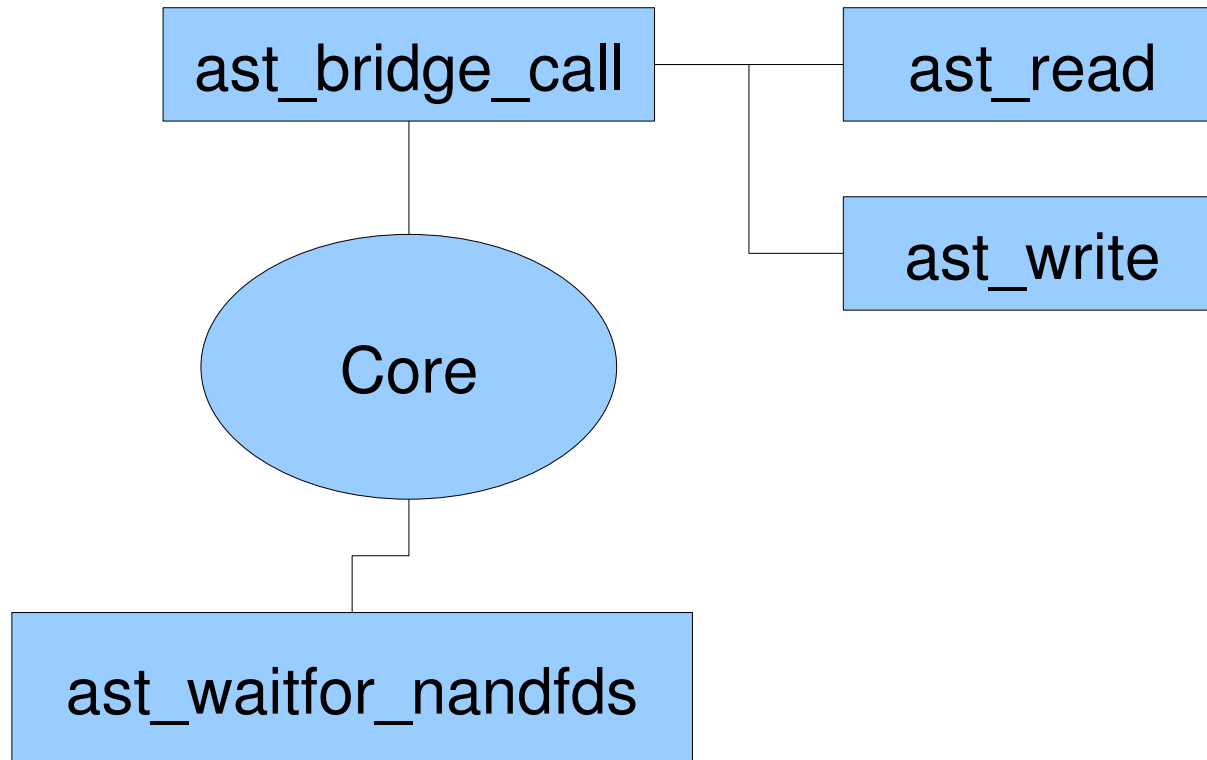
Bridging (Simple)

- Two (or more) channels
- Audio is exchanged between them
- Optional features are available

Bridging (API)

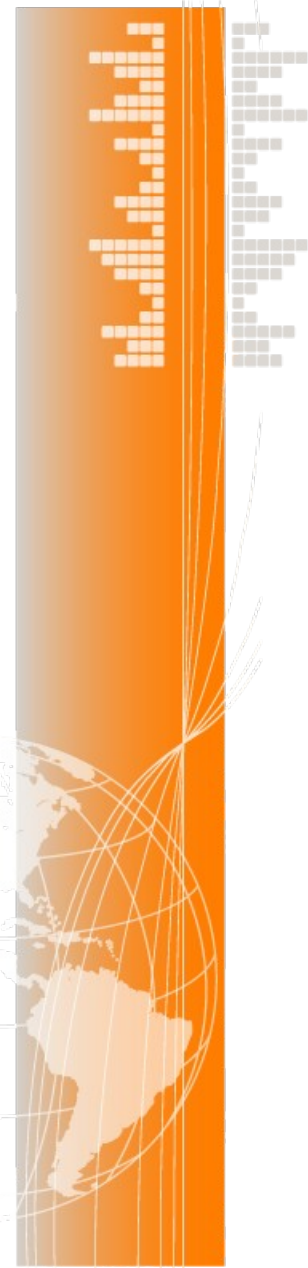
- `ast_bridge`
- `ast_waitfor_nandfds`
- `ast_read`
- `ast_write`

Bridging (Diagram)



Bridging (Internal)

- Channels are waited on
- If a frame is read from one it is written to the other
- If a channel hangs up the bridge is broken



Bridging (Future)

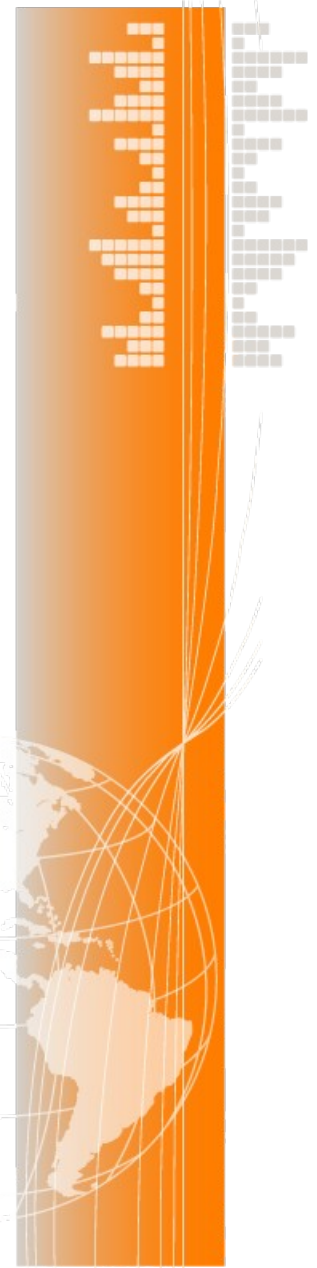
- `ast_bridge_new`
- `ast_bridge_join`
- `ast_bridge_impact`
- `ast_bridge_destroy`

Blind Transfer (Simple)

- Dialtone is provided
- Digits are collected
- Remote party is sent to new extension

Blind Transfer (API)

- `ast_app_dtget` (if Asterisk side)
- `ast_async_goto`



Blind Transfer (Internal)

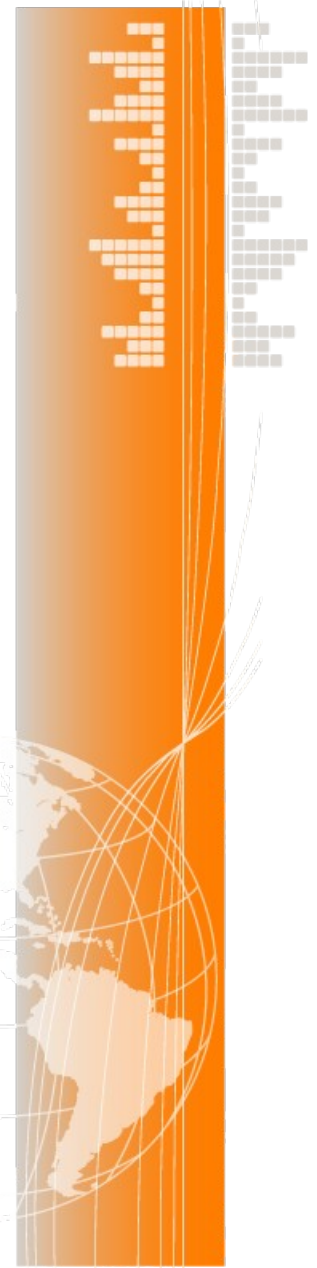
- If a PBX is running on the channel its context/extension/priority is changed
- If a PBX is NOT running it is started

Attended Transfer (Simple)

- Dialtone is provided
- Digits are collected
- New party is called
- Caller transfers new party to existing remote party

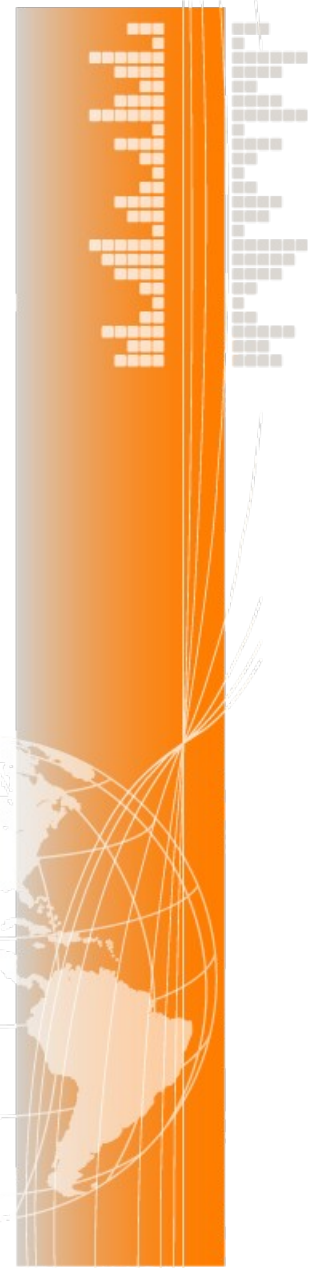
Attended Transfer (API)

- `ast_app_dtget` (if Asterisk side)
- `ast_channel_masquerade`



Masquerading

- Allows a channel to take the place of another at any time
- Mostly used by attended transfers
- Contents of each channel are exchanged
- One becomes a zombie, the other goes on normal



Questions?