What is LibWAT?

• First, lets go over two things:
  – Why use GSM
    • Your PBX is a fixed, it stays in your office, so why would you need a mobile technology with it?
  – GSM module interface basics
Why use GSM?

- **Cost Reduction**
  - In Europe and Latin America, a lot of providers have packages where cellular-to-cellular calls cost less than landline-to-cellular

- **Simple SMS Server**
  - Simple way to broadcast SMS to employees

- **Failover/Backup phone line**
  - If PSTN/POTS line goes down, can revert to GSM network.
GSM Module Interface Basics

• Hayes AT command set
  – Used by most GSM modules
  – Based on plain text simple commands
  – Common commands specified in 3GPP TS 27.007

• Message flow during a voice call
  – Outgoing call
    • Tx: ATD<number to dial>;
    • Rx: OK\r\n
  – Incoming call
    • Rx: RING\r\n    • Rx: +CLIP:"<caller ID>\r\n    • Tx: ATA\r\n
What is LibWAT?

What is LibWAT?

• Wireless AT Commands Library
• Protocol stack
  – Library that encodes/decodes Wireless AT commands.
• Similar role to LibPRI, LibSS7, OpenR2
  – Handles signalling and call control.
  – Controls when to answer a call, start media, hangup, etc…
Yes, there are several existing AT libraries for Asterisk, but...

- They are not included with Asterisk
  - Complex installation steps.
- Requires a patched version of Asterisk.
  - Users are limited to specific versions or releases of Asterisk.
- Some implementations even bundle the Asterisk source with their package.
- Each one of them support only specific GSM module vendors
  - If you want to switch hardware vendor, you would have to redo your dialplan and scripts.
  - Very unlikely for the Asterisk Community to adopt them.
Why we should have a generic AT Library

• More users = More Developers = More Features + More stable product
• Easier, straight forward installation
• All versions and releases of Asterisk supported
• Ability to switch hardware vendor without having to rewrite all your scripts.
• Installation would be simpler
• LibPRI and LibSS7 have been a success
Specifications/Goals when designing LibWAT

• Open Source
• Easy to add support for new AT commands and features
• Support for custom, vendor specific AT Commands
• AMI interface
• Easy maintenance and debugging
• Scalable
• Seamless integration with Asterisk
Architecture

Asterisk

chan_sip  chan_dahdi

LibPRI  LibSS7  LibWAT

DAHDI
Function send a command:
wat_status_t wat_cmd_enqueue(wat_span_t *span,
const char *command,
wat_cmd_response_func *cb,
void *cb_obj,
uint32_t timeout_ms);

**span**: pointer for span structure for that span
**command**: AT command to be transmitted
**cb**: Callback function when response is received
**cb_obj**: Callback object when response is received
**timeout_ms**: Timeout in milliseconds to wait for a response

Example:
wat_cmd_enqueue(span, "ATZ", wat_response_atz, NULL, 60000);
For each hardware vendor, we have provision for custom AT commands.

From telit.c, initialization function for Telit modules:

```c
wat_log_span(span, WAT_LOG_DEBUG, "Starting Telit module\n");

wat_cmd_enqueue(span, "AT#SELINT=2", wat_response_selint, NULL, command_timeout);

/* From Telit AT commands reference guide, page 105: Set AT#REGMODE=1
 * makes CREG behavior more conform to spec */
wat_cmd_enqueue(span, "AT#REGMODE=1", NULL, NULL, command_timeout);
wat_cmd_enqueue(span, "AT#DVI=1,1,0", wat_response_dvi, NULL, command_timeout);

/* Enable Echo cancellation */
wat_cmd_enqueue(span, "AT#SHFEC=1", NULL, NULL, command_timeout);
wat_cmd_enqueue(span, "AT#SHSEC=1", NULL, NULL, command_timeout);
```
AMI Interface for SMS

Sending a SMS:

```
try {
    $command = "Action: WATSendSms\n";
    $command .= "span:1\n";
    $command .= "To-Number:16476254279\n";
    $command .= "Content-type: text/plain; charset=UTF-8\n";
    $command .= "X-SMS-Class: ME-Specific\n";
    $command .= "Content: Hello World\n";
    $command .= "\n";

    $response = $ast->sendCommand($command);
    echo $response;
} catch (PEAR_Exception $e) {
    echo $e;
}
```
SMS Server deployment

- Implemented by one of our partners, Incuvox in Mexico for an electoral campaign.
- 12 Sangoma W400 (Total of 48 GSM channels)
- 100,000 SMS per hour
Currently Supported Features

• Voice Calls (/w progress detection)
• SMS
  – Plain Text
  – PDU mode (7-bit, UTF-8)
  – SMS Server
Download LibWAT

- Developers, download from Git Repository:
  - `git clone git://github.com/sangoma-dev/libwat.git`
- Users, latest stable release:
- Instructions:
Going forward…

- Future implementation
  - SIM Virtualization
  - Graphical SMS
- Asterisk community support
  - JIRA issue 19561
    - https://issues.asterisk.org/jira/browse/ASTERISK-19561
  - We more users and developers adopting LibWAT, we are hoping the Asterisk maintainers and Project Management will re-consider including LibWAT inside Asterisk.
Question and Answers

• Thank you

• If you would like to contribute or evaluate LibWAT, do not hesitate to contact us.
  – David Yat Sin <dyatsin@sangoma.com>