Sangoma Webinar Series
Telecom Tapping Solutions

F.Dickey / N.Corbic

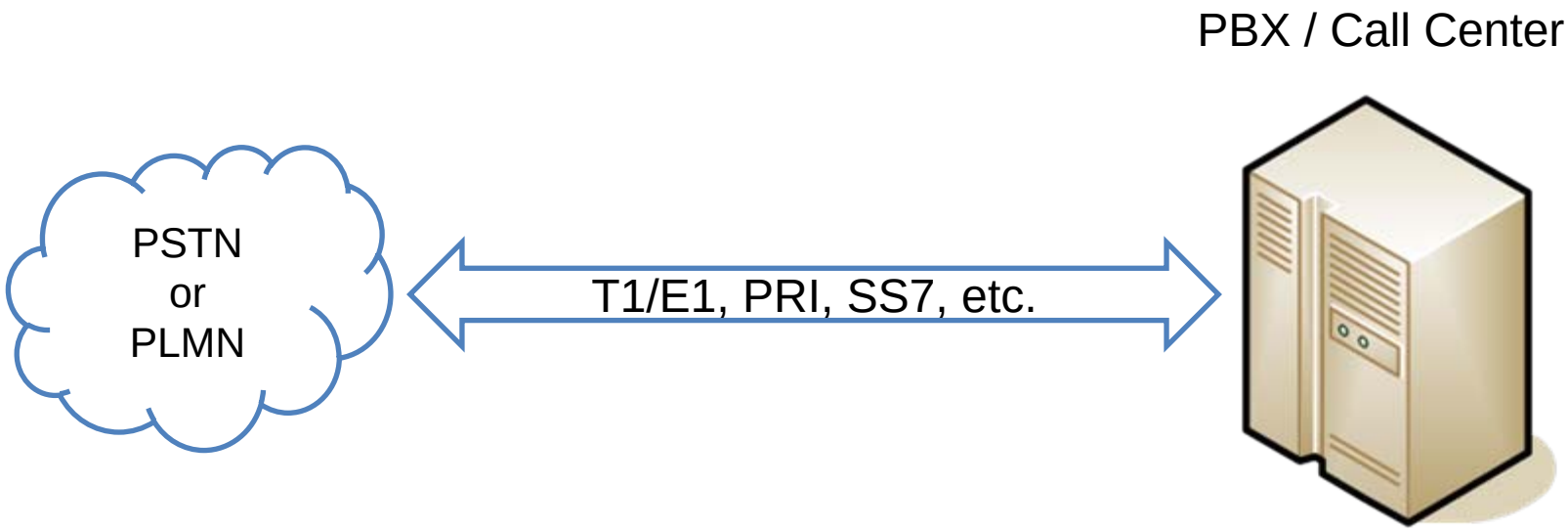
December 14, 2010

SANGOMA

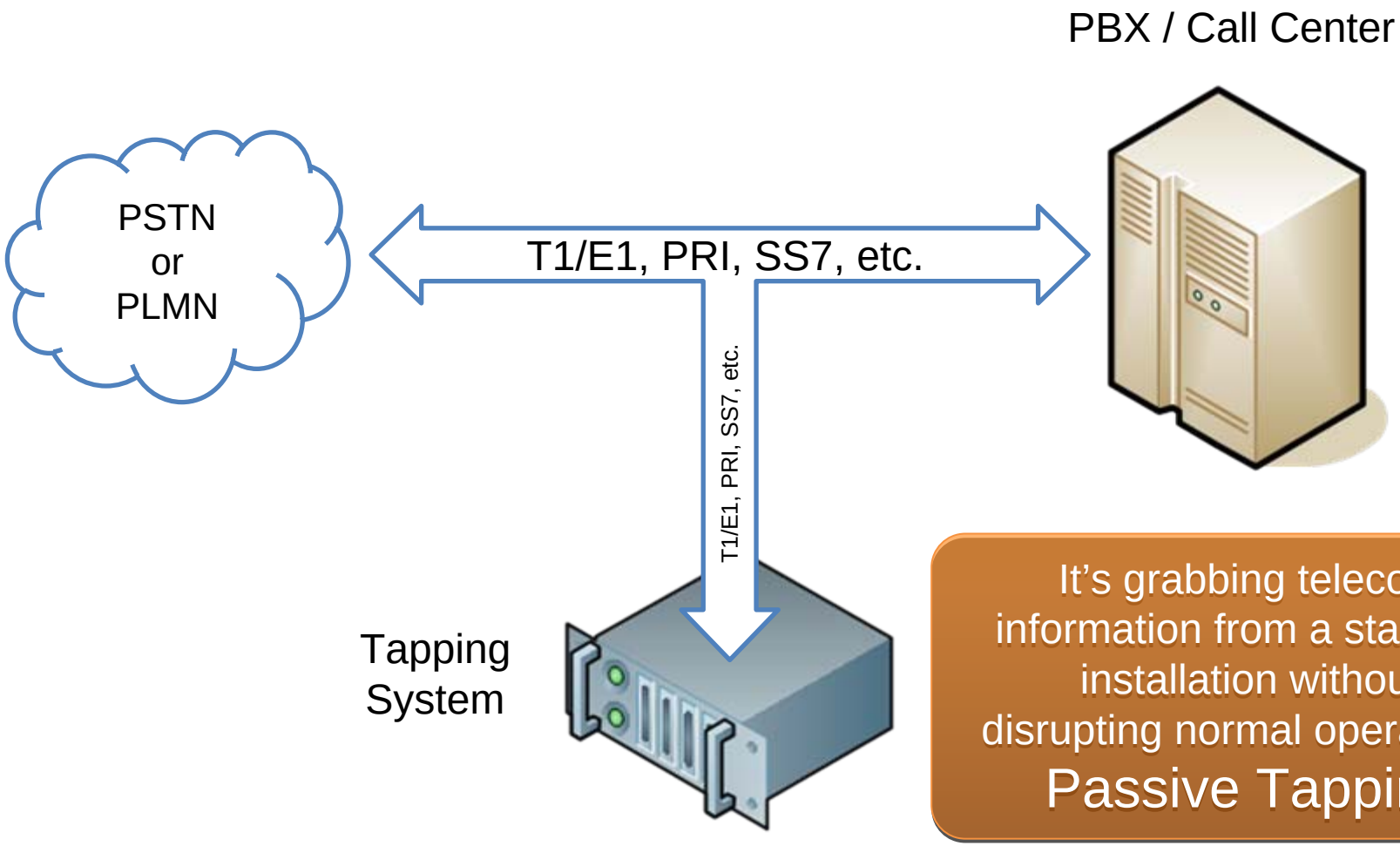
Agenda

- What is Tapping?
- What Telecom Applications would require Tapping?
- How can I do this with Sangoma?
  - Hardware products
  - API modes and configurations options
- Q&A
- Conclusion
- Winner Announcement

Start with a “normal” telecom set up



What is Tapping?

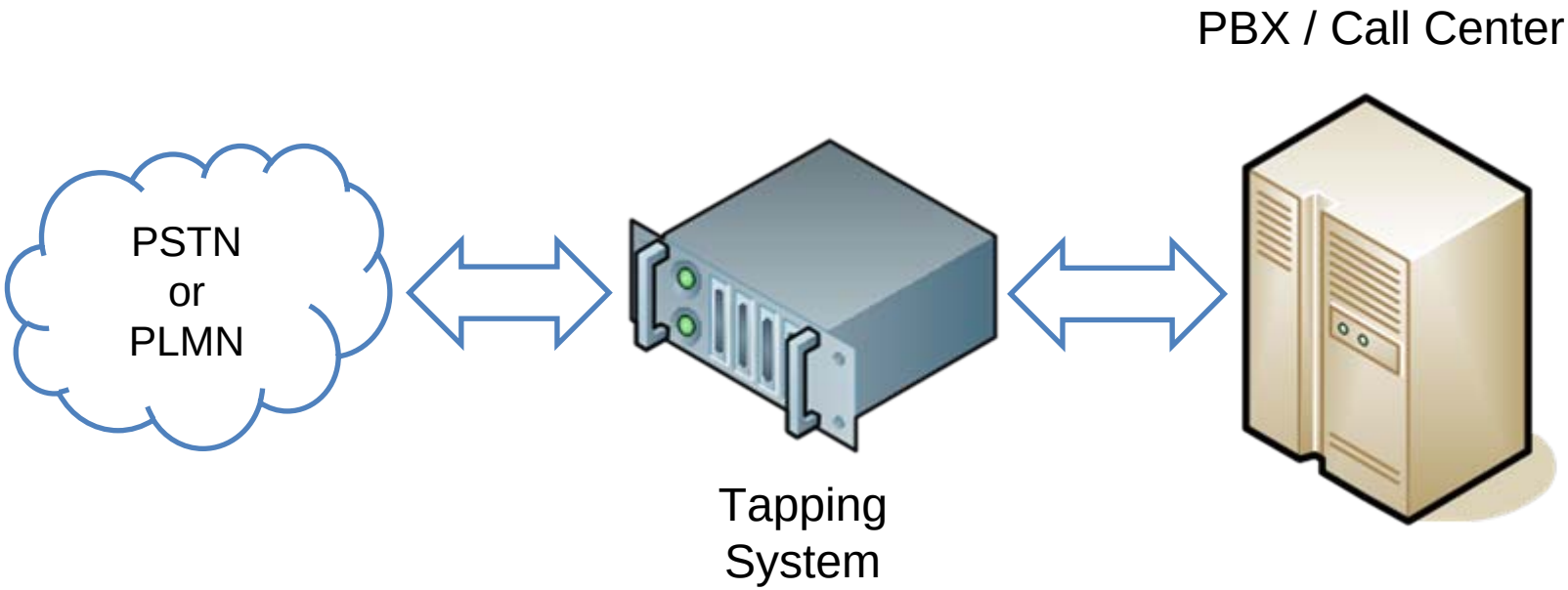


It's grabbing telecom information from a standard installation without disrupting normal operations:
Passive Tapping

Why would I want to do this?

- There is lot of useful information on telecom links:
  - Call Control information
  - Telecom Protocol information
  - Voice and Media
- It can be very useful to access this information for analysis
  - Fraud detection
  - Quality monitoring in call centers (recording)
  - Call logging for financial institutions
  - Security
  - Etc.

Is there another way to do this?

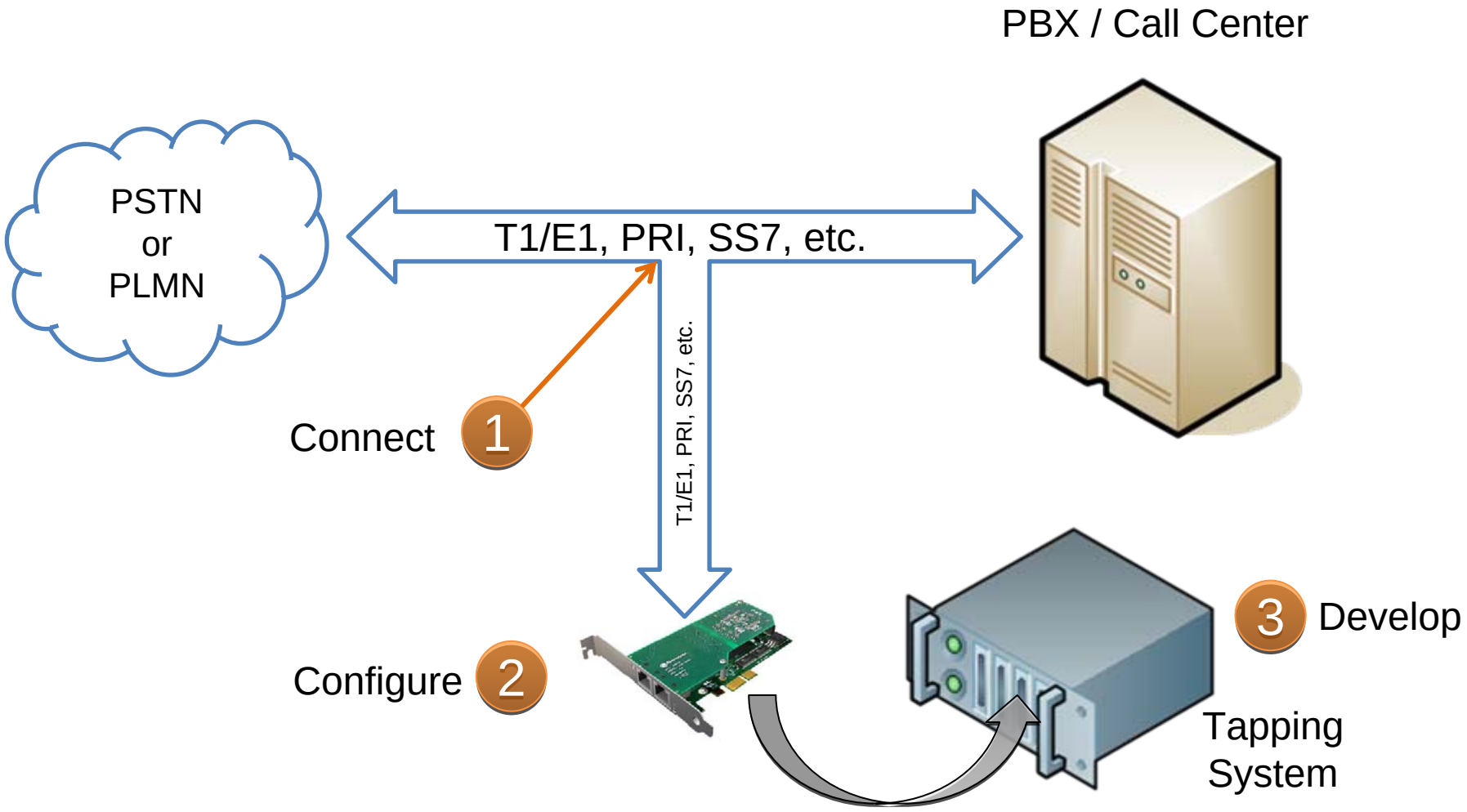


- Terminate and re-originate calls in-line
- Becomes single point of failure
- Passive Tapping is better

How do I get started with Passive Tap?

1. Create a passive tap connection
2. Set-up Sangoma Telephony card
3. Work your magic! (develop your app)

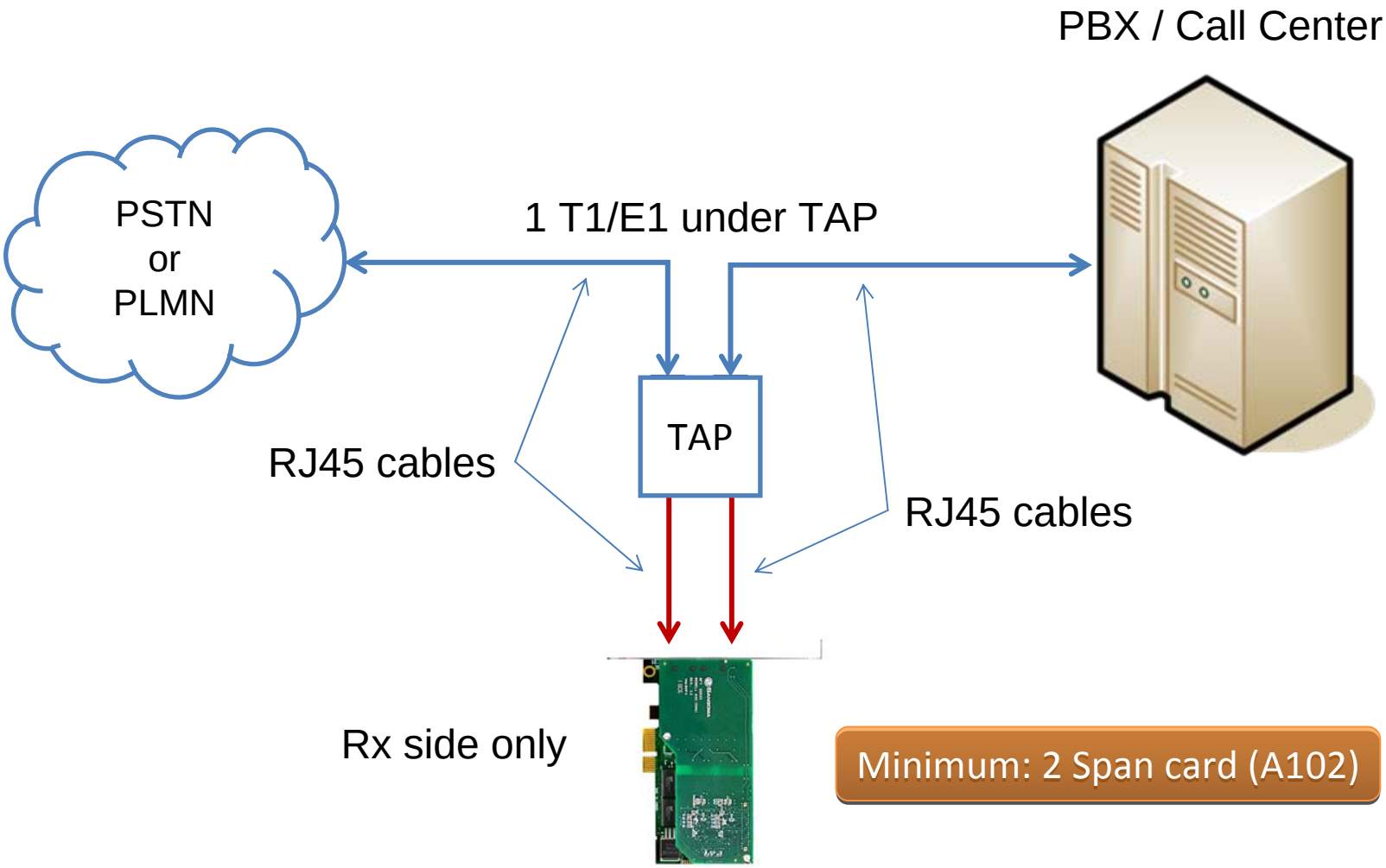
How do I get started (bis)?



1 Passive Tap Connection

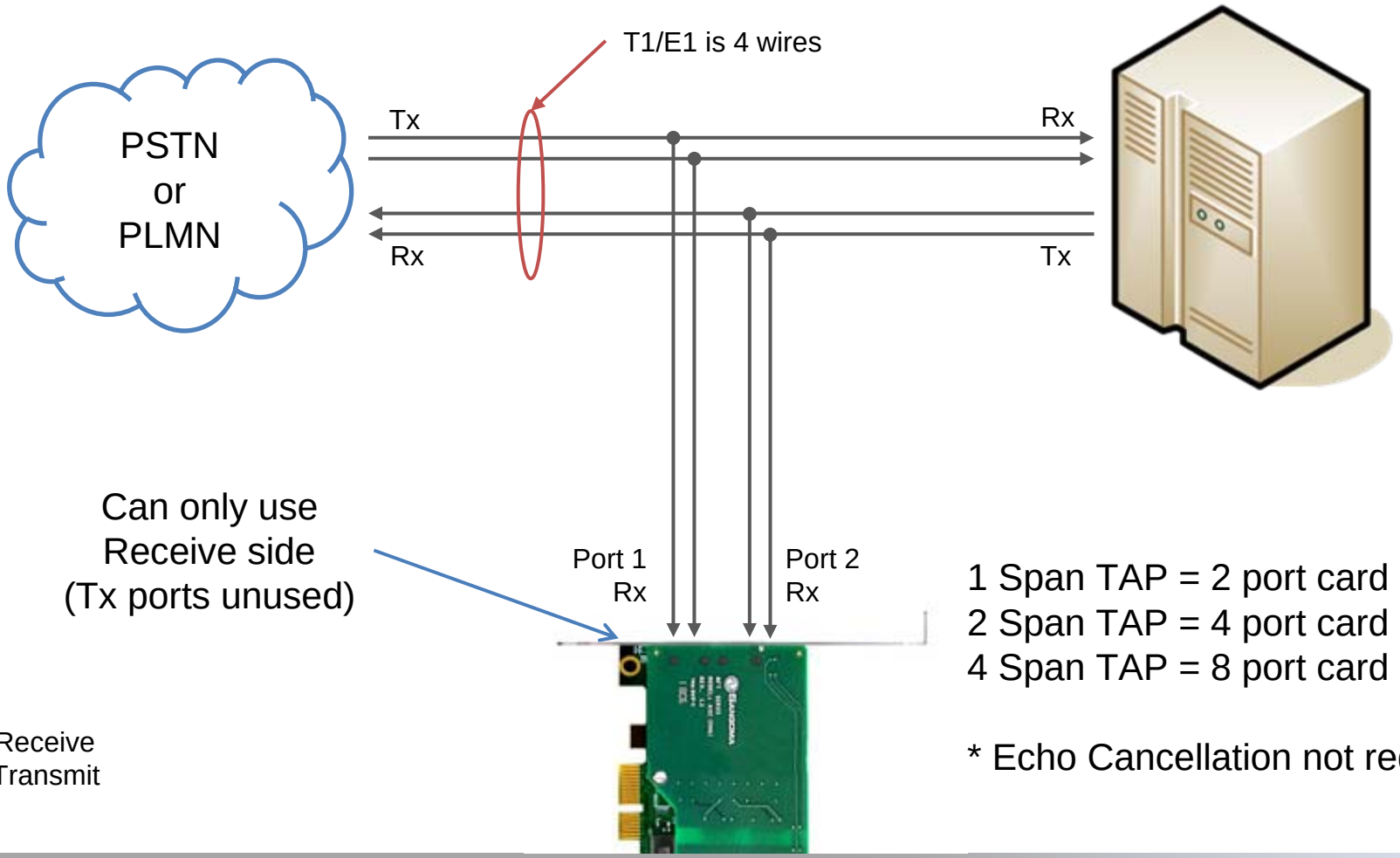
- To be passive, connection needs to have high impedance
- Some people build their own special cables with resistors
- Sangoma provides a tapping interconnect box that makes this easy for you
- Part number: TAP
- List Price: $150 USD
- 1 TAP box required for each T1/E1

Where does this TAP box fit?



Minimum: 2 Span card (A102)

Why do I need at least a 2 Span card (A102)?



1 Span TAP = 2 port card (A102)
2 Span TAP = 4 port card (A104)
4 Span TAP = 8 port card (A108)

* Echo Cancellation not required

Rx = Receive
Tx = Transmit

Sangoma Cards SKU and List Prices

SKU	Description	List Price ($USD)
A102	2 span card, PCI bus	$ 1000
A102E	2 span card, PCI Express bus	$ 1000
A104	4 span card, PCI bus	$ 1700
A104E	4 span card, PCI Express bus	$ 1700
A108	8 span card, PCI bus	$ 2900
A108E	8 span card, PCI Express bus	$ 2900

2 3 Sangoma Tapping Solutions

- SDK/API
  - LibSangoma + LibStelephony
  - FreeTDM
- Applications
  - SigTap App
  - RTP Tap
- Open Source Frameworks
  - Asterisk
  - FreeSWITCH

Wanpipe Configuration

- Interface in TDM_VOICE mode.
- TE_HIGHIMPEDANCE = YES
- Sample configs available:
  - http://wiki.sangoma.com/sangoma-tap-system

LibSangoma Telephony Base

User Library

LibSangoma
User Space Library

[config, start, stop, open,close,read,write,events]

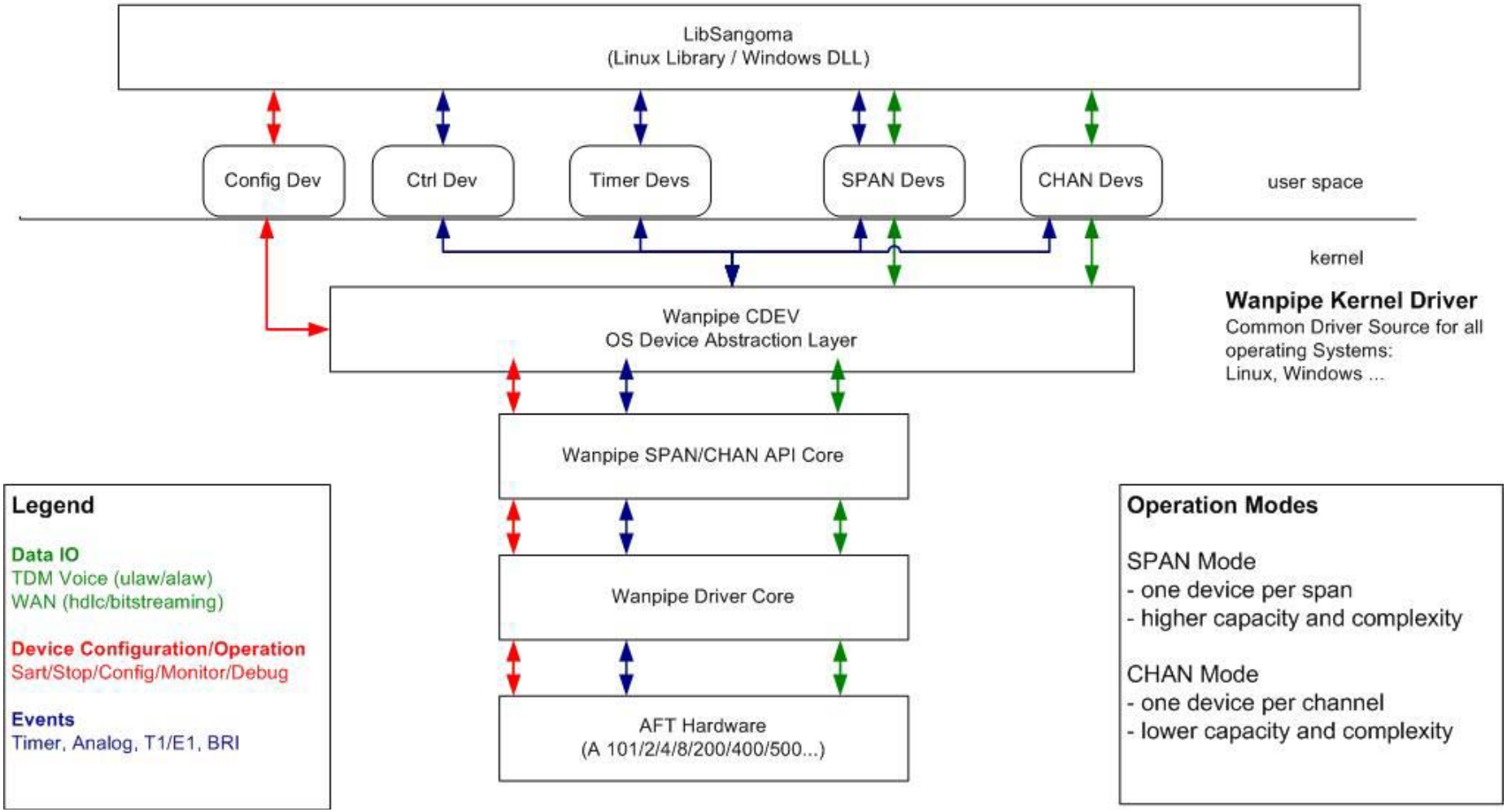
Kernel

WANPIPE TDM API
Kernel Interface

[T1,E1,BRI,Analog,Serial]

Sangoma Hardware

LibSangoma – more details



WANPIPE / LibSangoma + LibStelephony

- WANPIPE® Voice API
  - libsangoma is a device based, multi-threaded suite of kernel drivers and user space libraries that is used to build custom Voice applications over Sangoma TDM Voice hardware, on both Linux & Windows.
- LibSangoma:
  - Is a user space library (.so/.dll)
  - Abstracts Linux and Windows system calls into a OS agnostic API
  - Provides a common API for all Sangoma hardware.
  - Is a very low level “RAW” API. It has not logic or state machines.
  - Provides greatest flexibility for a developer, consequently its hardest to use.
  - Distributed under BSD license
- LibStelephony
  - Is a helper user space library (.so/.dll)
  - implements voice processing: callerid, dtmf
  - Implements PRI decoding

LibSangoma Components

- Configuration Functions
  - configure T1/E1, analog, BRI ports before they are started or restarted
- System Operation Functions
  - used to start, stop, restart Sangoma tdm ports
- IO Operation Functions
  - used to open, close, read, write,poll on bchan or dchan devices
  - set events that relate to IO or call setup
    - callerid, on-hook, off-hook, rbs etc…
- Event Functions
  - used to read events from bchan or dchan devices.
  - Used to enable or disable hw events (DTMF, Fax events …)
- Statistic Functions
  - used to read hw and driver statistics.
  - Operational, Hardware, Error, IO, statistics

LibSangoma I/O modes

- SPAN Mode
  - Single device per T1/E1 span.
  - Data from a single span is passed up to a user as a single chunk of data.
  - User has the responsibility to multiplex and de-multiplex data per channel.
  - Better performance on higher densities.
  - Harder to develop on.
- CHAN mode
  - Device per T1/E1 channel
  - Highly efficient interrupt logic to minimize number of interrupts.
  - High kernel context switching due to large number of devices.
  - Easer to develop on. Easily fits into a multi-thread applications.
  - Default behavior for Zaptel/DAHDI, FreeSWITCH, SMG

LibSangoma Tapping Sample App

- Libsangoma is distributed as part of Wanpipe Driver Release.
- Libsangoma Location:
  - wanpipe-<ver>/api/libsangoma
- C Sample:
  - wanpipe-<ver>/api/sample_c
- C++ Sample:
  - wanpipe-<ver>/api/sample_cpp

- Each sample API will can open a b-chan or a d-chan and start reading data.
- Its up to the user to mix the voice from 2 b-chan
- Its up to the user to decode d-chan frames (or use libstelephony)

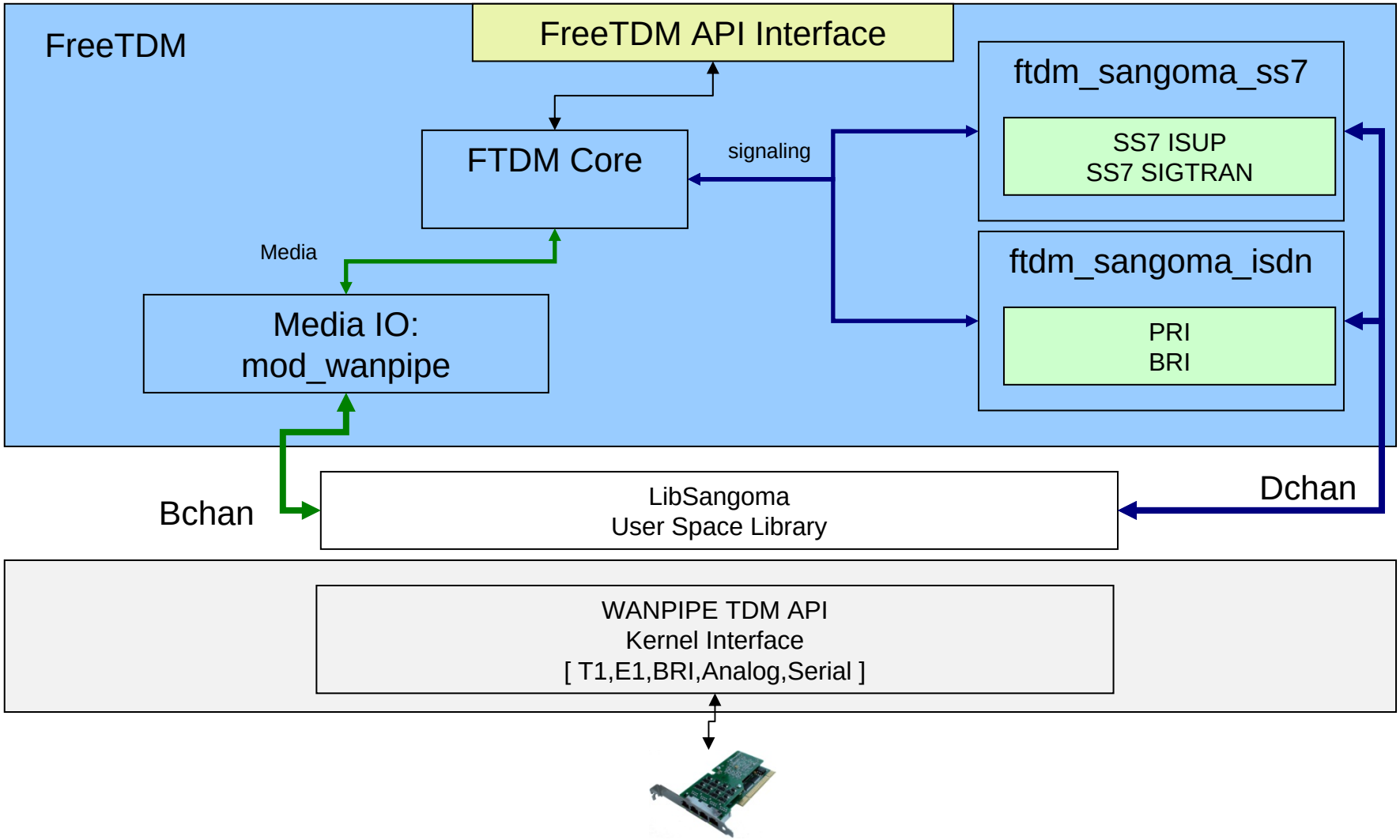
LibSangoma Reference

- Reference:
  - http://wiki.sangoma.com/wanpipe-api
- Doxygen Documentation:
  - http://docs.sangoma.com/doxygen/libsangoma/html/files.html

FreeTDM

- An open source TDM Framework. It builds on top of Libsangoma functionality and includes full TDM protocols and state machines as well as voice processing
- TDM Endpoint for FreeSWTICH project
- A user space library (.so/.dll)
- Abstracts Linux and Windows system calls into a OS agnostic API
- Provides a common Voice IO and Signaling API for all Sangoma hardware.
- Incorporates Sangoma Telco Grade Stacks:
  - SS7,PRI,BRI,Analog,R2
- Provides a common Signaling API to above mentioned stacks.
- Provides Signaling Media processing: CallerID, FSK, E&M Wink
- Provides IO Media processing: DTMF, Fax detect
- Sangoma IO module is based on Libsangoma library: CHAN mode.
- Distributed under BSD license

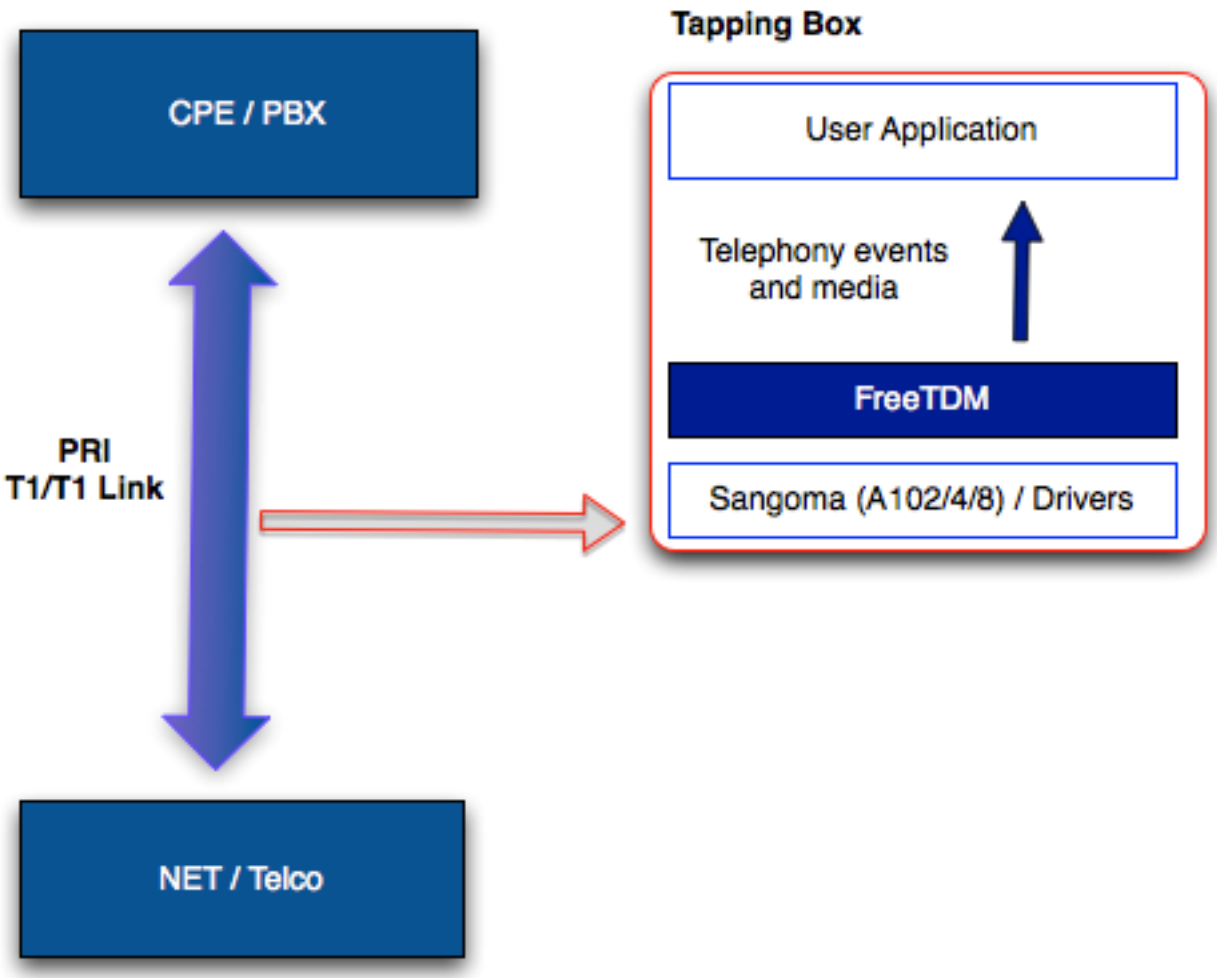
FreeTDM API


FreeTDM
FreeTDM API Interface
FTDM Core
signaling
ftdm_sangoma_ss7
SS7 ISUP
SS7 SIGTRAN
ftdm_sangoma_isdn
PRI
BRI
Media
Media IO:
mod_wanpipe
Bchan
LibSangoma
User Space Library
Dchan
WANPIPE TDM API
Kernel Interface
[T1,E1,BRI,Analog,Serial]

FreeTDM Components

- I/O Operation Functions
  - used to open, close, read, write, poll on bchan or dchan devices
  - set events that relate to IO or call setup: callerid, on-hook, off-hook, rbs etc… call recording functions (start, stop…)
- Signaling Functions**
  - used to place call, clear call, set channel into maintenance.
  - signaling events are used to receive signaling events such as progress, proceed etc…
- Event Functions
  - used to read events from bchan or dchan devices.
  - Used to enable or disable hw events. (DTMF, Fax events …)
- Statistic Functions
  - used to read hw and driver statistics.
- Operational, Hardware, Error, IO, statistics
- Configuration Functions
  - not supported: one would use libsangoma directly to configure the TDM ports.
- System Operation Functions
  - not supported: one would use libsangoma directoy to start, stop, restart Sangoma tdm ports.

FreeTDM Tapping Module



FreeTDM PRI Tapping Module

- Easier API in C for PRI tapping.
- You configure your spans with “pritap” signaling.
- Calls are reported through regular SIGEVENT_ messages.
- Using I/O FreeTDM API you can access the mixed stream.
  - ftdm_channel_read(ftdmchan) returns the stream already mixed.

FreeTDM PRI Tapping Module

- Uses passive version of libpri for message decoding.
  - http://svn.digium.com/svn/libpri/team/moy/tap-1.4/
- Decodes IE's on SETUP, PROCEED, ALERTING, CONNECT, DISCONNECT, etc
- Planning to move to independent decoder to drop dependency.
- Configure FreeTDM with –with-pritap to enable ftmod_pritap.so.

FreeTDM Configuration

- Regular T1/E1 configuration parameters for 2 spans (or more).

```
[span wanpipe tap1]
trunk_type => T1
b-channel => 1:1-23
d-channel => 1:24

[span wanpipe tap2]
trunk_type => T1
b-channel => 2:1-23
d-channel => 2:24
```

FreeTDM Reference

- FreeTDM is distributed as stand alone product.
- Reference:
  - http://wiki.sangoma.com/wanpipe-api-freetdm
  - http://wiki.freeswitch.org/wiki/FreeTDM
- Doxygen Documentation:
  - http://docs.sangoma.com/doxygen/freetdm/

2 3 Sangoma Tapping Solutions

- SDK/API
  - LibSangoma + LibStelephony
  - FreeTDM
- Applications
  - SigTap App
  - RTP Tap
- Open Source Frameworks
  - Asterisk
  - FreeSWITCH

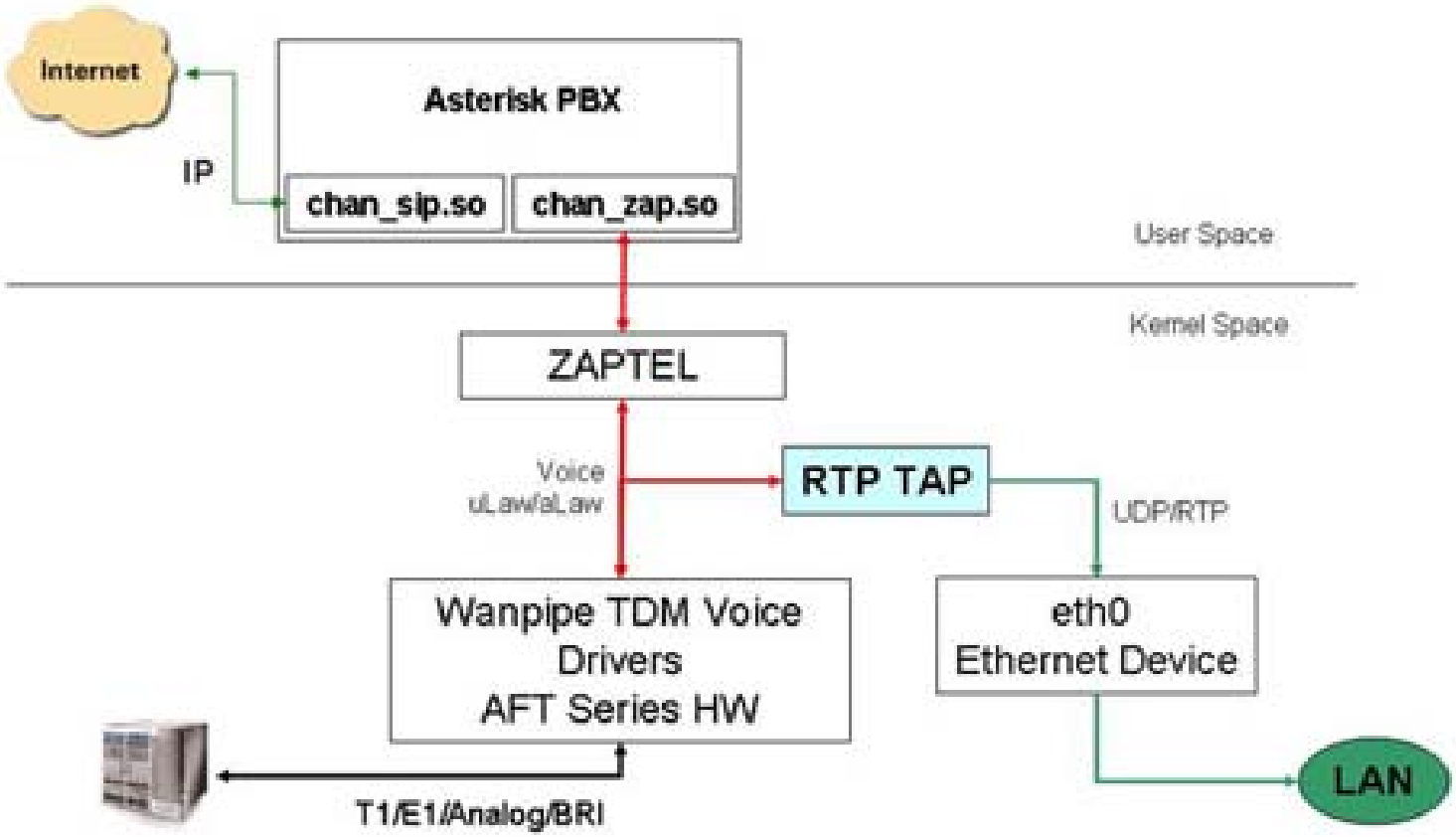
SigTap Application

- SigTap is signaling tapping application written on top of libsangoma.
  - Application written in C++
  - Build on top of libsangoma API
  - Runs on Linux & Windows
- Purpose of SigTap is to tap signaling channels NOT VOICE.

SigTap Features

- Auto configures T1/E1 cards into SPAN mode.
- Auto detects T1/E1 configuration.
- Keeps trying all possibilities until the T1/E1 line becomes connected.
- Once line becomes connected it starts receiving JUMBO (16K to 64K) frames.
- Its developers responsibility to de-multiplex the JUMBO frames and auto-detect where the signaling channels are.
- Its developers responsibility to decode and handle signaling data.
- Highly scalable: Up to 64E1s on single machine.

RTP TAP


Internet
IP
Asterisk PBX
chan_sip.so
chan_zap.so
User Space
Kernel Space
ZAPTEL
Voice
uLaw/aLaw
RTP TAP
UDP/RTP
Wanpipe TDM Voice
Drivers
AFT Series HW
eth0
Ethernet Device
LAN
T1/E1/Analog/BRI

RTP TAP Features

- The "Wanpipe Voice RTP TAP" is used to tap voice channels at the driver/kernel level during Asterisk-Dahdi or TDM API operation mode.
- All tapped voice channels, in Wanpipe driver, are encapsulated in UDP and RTP headers and transmitted over the neighboring Ethernet card, within the kernel.
- Since data never leaves the kernel the tapping feature is super efficient. Tapping 4 E1s worth of voice takes only 2% of overall system load.
- RTP TAP is a feature of Wanpipe Linux Kernel Driver
- Can be enabled by adding extra configuration to existing wanpipe1.conf file.
- Used to tap existing Asterisk/FreeSWITCH applications.
- Non intrusive way to add tapping/recording to an Asterisk/FreeSWITCH system.

RTP TAP Operations

- RTP TAP is a feature of WANPIPE Linux Kernel Driver
- RTP TAP configuration is part of WANPIPE configuration files /etc/wanpipe/wanpipe1.conf
- RTP TAP feature works with Asterisk+DAHDI on T1/E1/Analog devices.
- Default Configuration Parameters:
  - RTP_TAP_IP=192.168.1.240 # Remote Server Address
  - RTP_TAP_MAC=00:09:1B:89:61:BC # Remote Server MAC
  - RTP_TAP_PORT=9000 # Starting UDP Port
  - RTP_TAP_SAMPLE=100 # RTP packet lenting in ms (100ms * 8 = 800 bytes)
  - RTP_TAP_DEV=eth0 # Local eth interface to send UDP data
- RTP TAP is automatically started by the driver when a call is established on a channel.
- All Rx & Tx data is concatenated into RTP_TAP_SAMPLE number of bytes and pre-pended with an IP/UDP/RTP header and sent over adjacent ethernet device RTP_TAP_DEV.
- Rx and Tx streams are transmitted separately.
- Rx stream UDP Port = RTP_TAP_PORT + channel number + 32*(span_number-1)
  - eg: span 1 channel 5 will be transmitted on UDP: 9005 (span=1 chan=5)
- Tx stream UDP Port = RTP_TAP_PORT + channel number + 2000 + 32*(span_number-1)
  - eg: span 1 channel 5 will be transmitted on UDP: 11005 (span=1 chan=5)

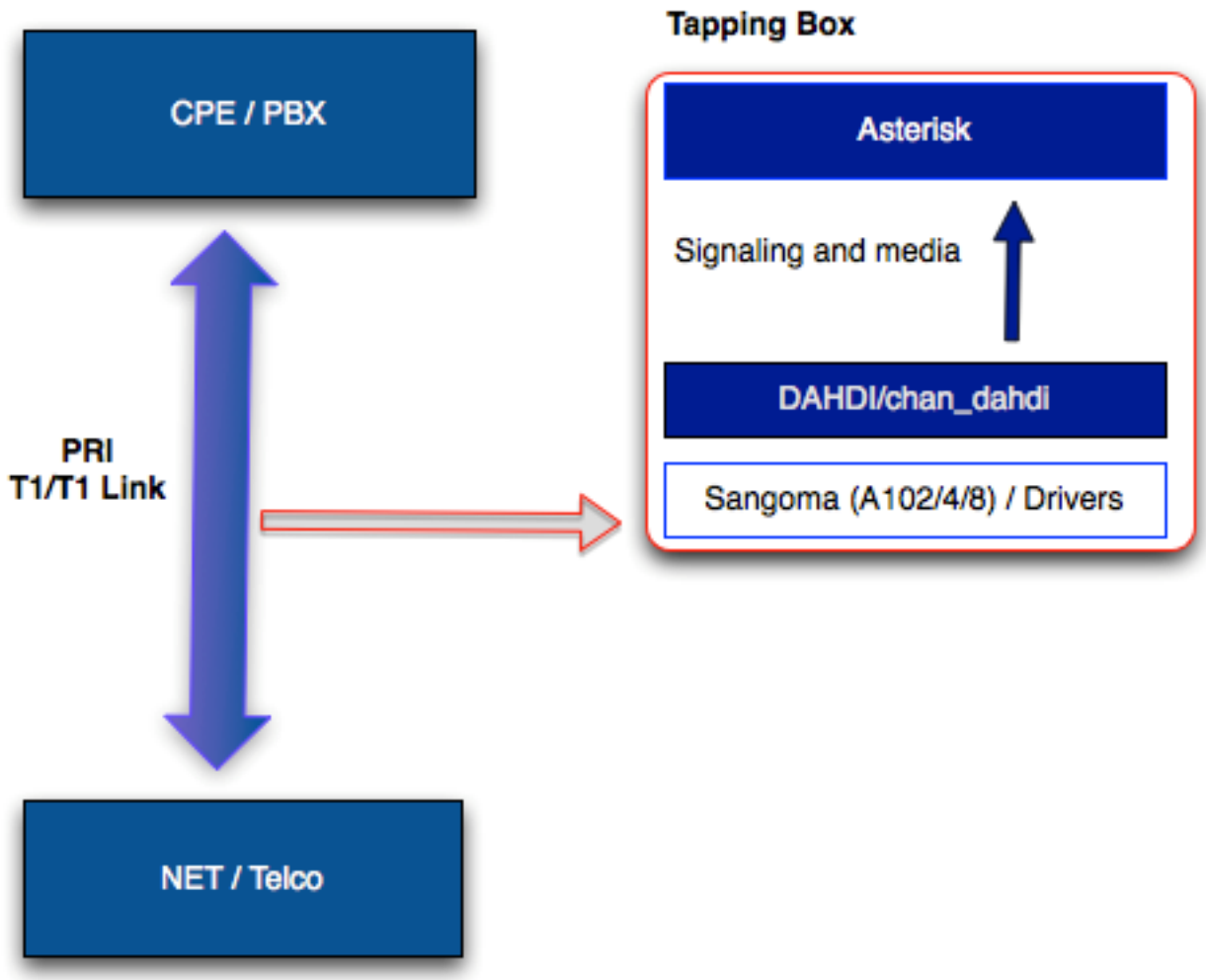
RTP TAP Reference

- RTP TAP Reference
  - http://wiki.sangoma.com/wanpipe-voice-rtp-tap
- Third Party Solution Partner
  - Orecx
  - Orecx has developed an Asterisk Call Recording Solution with Sangoma RTP TAP Technology

2 3 Sangoma Tapping Solutions

- SDK/API
  - LibSangoma + LibStelephony
  - FreeTDM
- Applications
  - SigTap App
  - RTP Tap
- Open Source Frameworks
  - Asterisk
  - FreeSWITCH

Asterisk PRI Tapping



Asterisk PRI Tapping

- chan_dahdi matches signaling messages from different spans.
- chan_dahdi creates an Asterisk channel and provide the mixed audio to the Asterisk core.
- Asterisk see the call as a regular incoming call.
  - Smells like a call.
  - Tastes like a call
  - But brother, it ain’t a regular call!

Asterisk PRI Tapping

- You can use regular Asterisk dial plan logic to do recording, logging or execute any other supported Asterisk application on the tapped call.
  - `exten => _X.,1,Answer()`
  - `exten => _X.,n,Record(…)`
- Other applications, like ChanSpy() can be used for live monitoring.
- Any application doing only media writing, won't fail, but won't do anything useful either.
- Any call control operation (Answer, Ring, Hang Up) is local only, does not affect tapped call, since there is no Tx enabled.

Asterisk PRI Tapping Installation

- Uses passive version of libpri for message decoding.
  - http://svn.digium.com/svn/libpri/team/moy/tap-1.4/
- Decodes IE's on SETUP, PROCEED, ALERTING, CONNECT, DISCONNECT, etc
- chan_dahdi required changes to drop data, match peer tapping spans and mix audio.
  - http://svn.digium.com/svn/asterisk/team/moy/dahdi-tap-1.6.2
  - http://svn.digium.com/svn/asterisk/team/moy/dahdi-tap-trunk
- Need your feedback to integrate into Asterisk trunk!

DAHDI Configuration (system.conf)

- Regular T1/E1 configuration parameters for 2 spans (or more).
- Remember you need 2 spans per T1/E1 link.
- Sample configuration for E1 tapping:

```
# Rx span
span=1,1,0,ccs,hdb3
bchan=1-15,17-31
hardhdlc=16

# Tx span
span=2,2,0,ccs,hdb3
bchan=32-46,48-62
hardhdlc=47
```

Asterisk PRI tapping Configuration

- chan_dahdi.conf

switchtype=national

context=from-tapped-line

signalling=pri_cpe

passive=yes

channel => 1-15,17-31

channel => 32-46,48-62

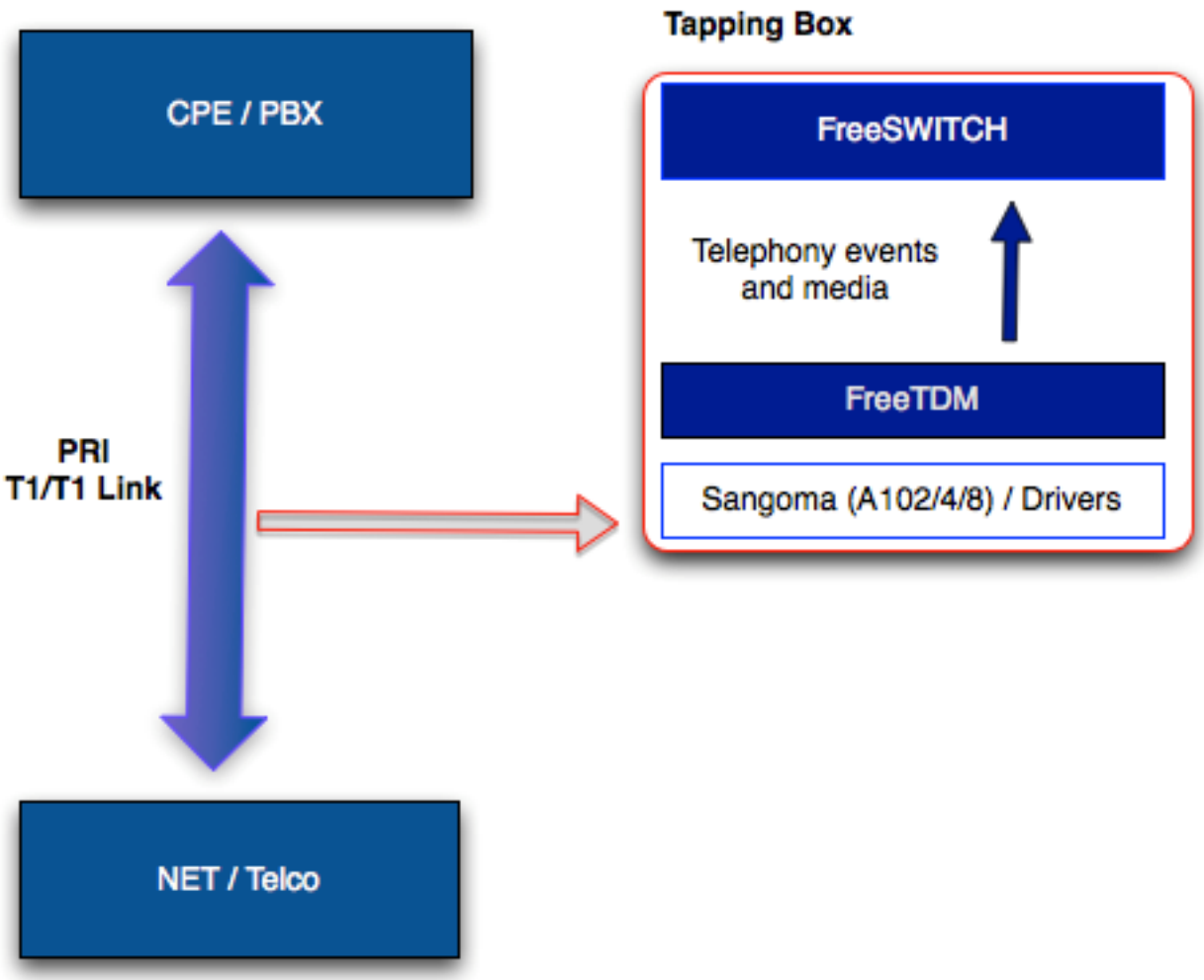
Asterisk PRI Tapping Key Points

- Two spans are required per tapped trunk.
- No outgoing calls can be placed in those circuits.
- No media writing (only media reading).
- Restarting the tapping server or Asterisk is safe.
- Exported PRI fields:
  - Caller Name
  - Caller Number
  - Called Number

Asterisk Tapping Conclusion

- You can build now a passive call recorder/logger easily.
- Tapped system can be any PRI switch/telco.
- Available in API mode or using standard Asterisk/DADHI integration.
- Extensible through regular dial plan logic, AGI scripts etc.

FreeSWITCH PRI tapping



FreeSWITCH PRI tapping

- FreeSWITCH acts as an FreeTDM application. No modifications at all needed in FreeSWITCH.
- FreeTDM reports tapped calls to FreeSWITCH as regular incoming calls.
- You use FreeSWITCH dial plan to do recording, logging or any other supported FreeSWITCH application on the tapped call.
  - <action application="record" data="….">
- FreeTDM drops all tx packets coming from FreeSWITCH.

FreeSWITCH PRI tapping Configuration

- XML configuration in autoload_configs/freetdm.conf.xml

```
<configuration name="freetdm.conf" description="FreeTDM Configuration">
  <pritap_spans>

    <span name="tap1">
      <param name="peerspan" value="tap2"/>
      <param name="dialplan" value="XML"/>
      <param name="context" value="default"/>
    </span>

    <span name="tap2">
      <param name="peerspan" value="tap1"/>
      <param name="dialplan" value="XML"/>
      <param name="context" value="default"/>
    </span>

  </pritap_spans>
</configuration>
```

FreeSWITCH PRI Conclusion

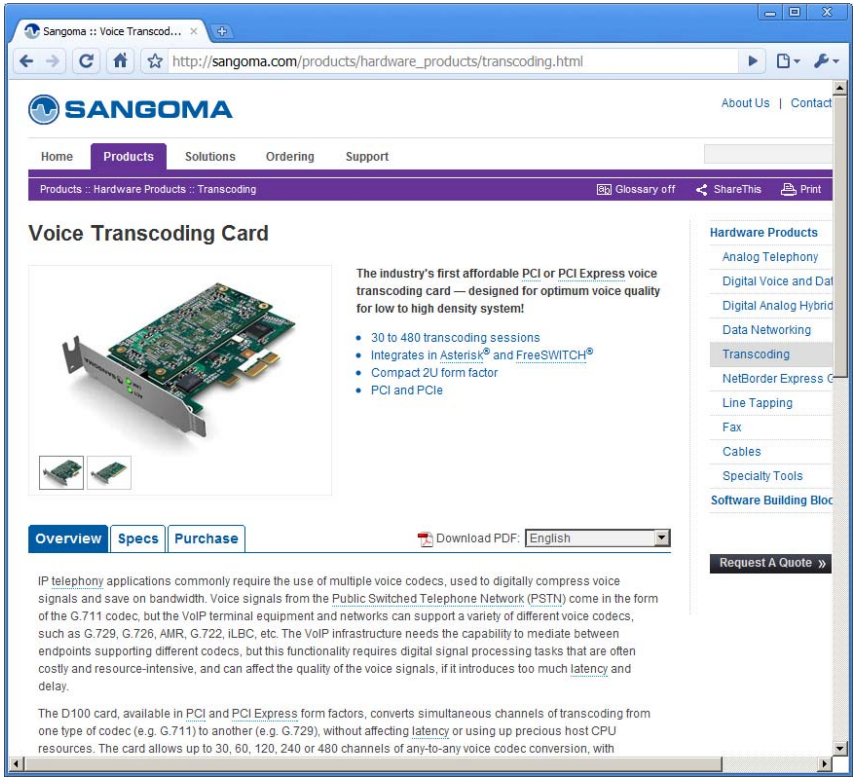
- You can build now a passive call recorder/logger easily.
- Tapped system can be any PRI switch/telco.
- Available in API mode or using standard FreeSWITCH/FreeTDM integration.
- Extensible through regular dial plan logic (XML, LUA etc).

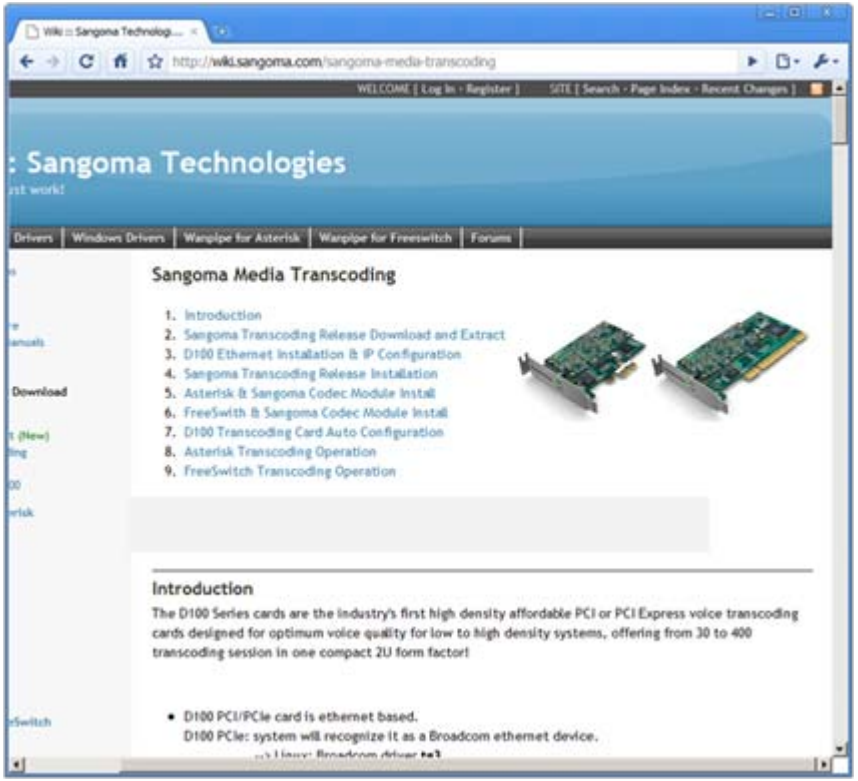
2 3 Sangoma Tapping Solutions

- SDK/API
  - LibSangoma + LibStelephony
  - FreeTDM
- Applications
  - SigTap App
  - RTP Tap
- Open Source Frameworks
  - Asterisk
  - FreeSWITCH

For more information:

- sangoma.com
- wiki.sangoma.com





Conclusion

- Passive Tapping useful for a lot of applications
- Need to set-up passive / high impedance connection
  - Tapping box from Sangoma
- Use Sangoma Boards for Tapping Application
- Several options for maximum flexibility
  - API/SDK
  - Apps
  - Open Source Frameworks

Thank You!

Sales: +1.905.474.1990
sales@sangoma.com

Frederic Dickey
Marketing and Product Management
fdickey@sangoma.com

Nenad Corbic
Software R&D
ncorbic@sangoma.com