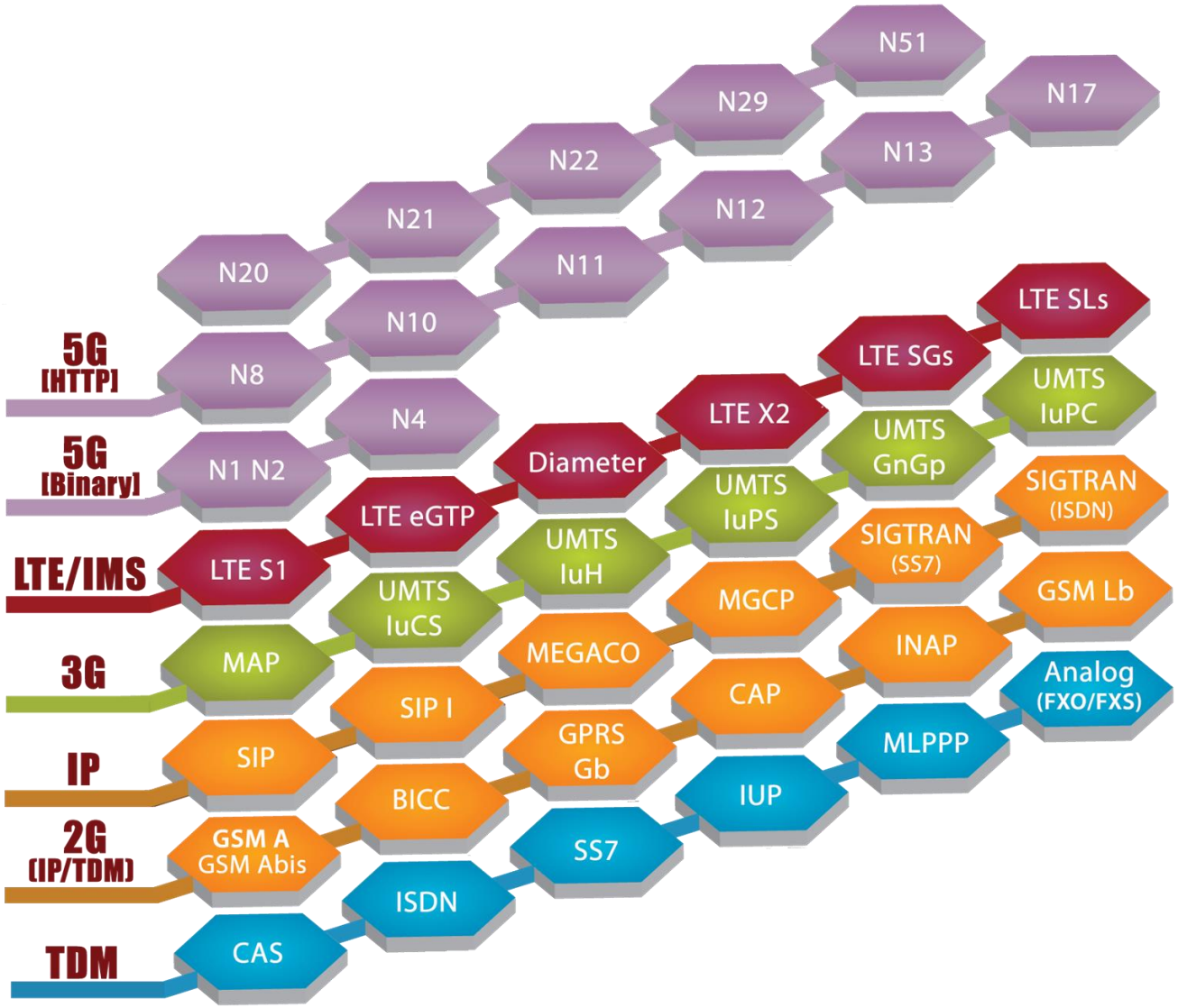
Signaling and Traffic Simulation using MAPS

GL Communications Inc.
818 West Diamond Avenue - Third Floor, Gaithersburg, MD 20878
Phone: (301) 670-4784 Fax: (301) 670-9187 Email: info@gl.com
Website: https://www.gl.com

About MAPS™

- **MAPS** stands for **M**essage **A**utomation and **P**rotocol **S**imulation
- It is a generic framework for the generation of telecommunications protocol messages and transmission of bearer traffic
- **MAPS™** is built on a proprietary scripting language developed by GL Communications
- All **MAPS™** products come with out-of-the-box scripts that act as fully functional state machines for the relevant protocol



GL Communications

About MAPS™ (Contd.)

- **Scripts:** Scripts act as the state-machine, or engine for a given call. The logic of what messages to send when is all contained in a script
- **Messages:** MAPS has an inventory of generic Message Templates (ex: Invite.txt) which it loads from the hard drive when transmitting an actual message. Messages are completely customizable
- **Profiles:** Scripts and Messages are kept as generic as possible. Specific information (ex: Contact = 12345@sip.carrier.com) about a call is sourced from .xml profiles

Basic Requirements for Emulation

- **Message Templates**
  - The message templates are nothing but structure of message stored in particular file format. e.g.: SS7 Protocol suite message template will have “. HDL” format
- **A ‘Script’**
  - To send and receive these messages between two nodes and take appropriate actions for a particular message
- **An ‘Import’ mechanism**
  - A mechanism for reading the contents of the message template and replacing the Key Identifier with the value given by the user (or some other means) at the run time. This process of inserting the user values into the message template before sending is called “Import”

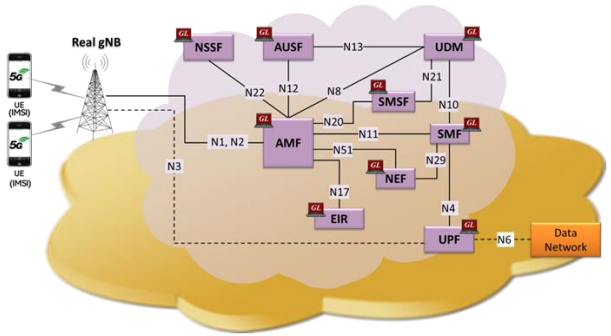
Basic Requirements for Emulation (Contd.)

- **An ‘Export’ mechanism**
  - A mechanism to extracting Key Identifier values from the received response and store for the future use (in the same call scenario) is called “Export” (This exported value can also be imported to message template in future)
- **A ‘Profile’ file**
  - Once the Key Identifiers are identified for all the message templates in a call scenario, required values are configured for these Key Identifiers in a file called Profile

GL Communications

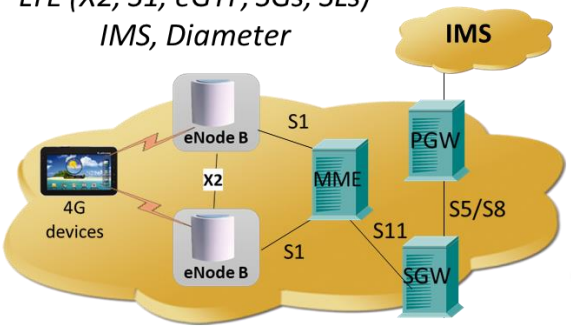
Supported Protocols / Interfaces

5G (N1N2, N4, N8, N10, N11, N12, N13, N14, N17, N20, N21, N22, 29, 51)



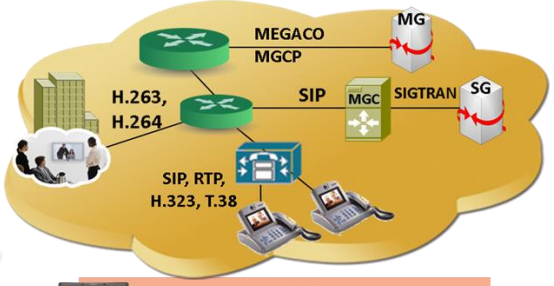
GL 5G NR & Core Access

LTE (X2, S1, eGTP, SGs, SLs)
IMS, Diameter

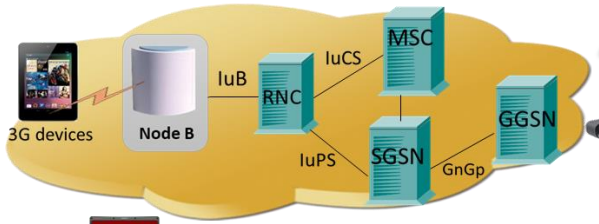


GL 4G Access (E-RAN)

SIP, Megaco, MGCP, SIP-I, SIGTRAN (ISUP, ISDN)

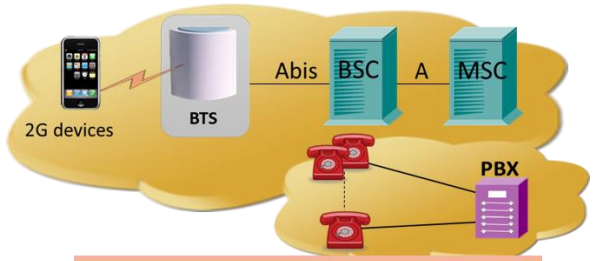


GL VoIP (Packet Network)



GL 3G Access (UTRAN)

UMTS (IuH, IuCS, IuPS, GnGp) GPRS



GL *2G Access (GERAN) TDM/PSTN Network*

CAS, ISDN, MLPPP, SS7, IUP, MAP, CAMEL, INAP , GSM (A, Abis)

Signaling, Traffic Generation and Analysis

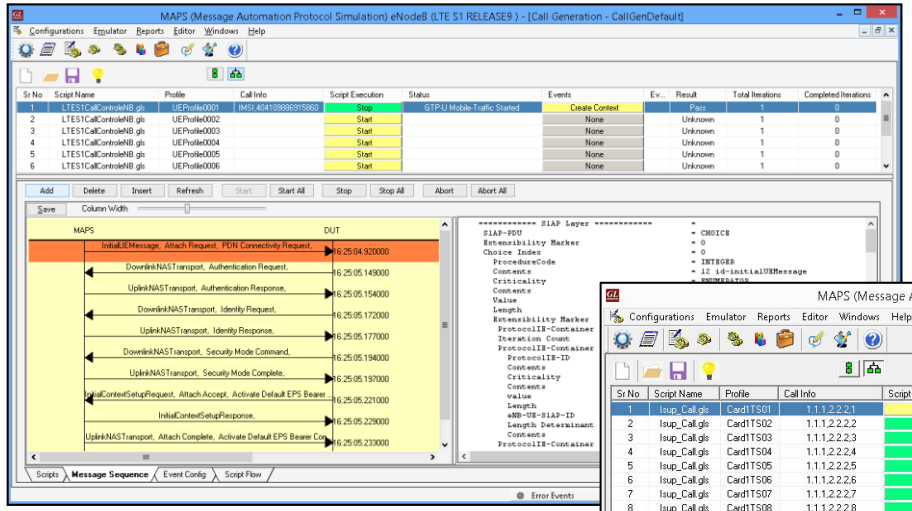
GL MAPS™ 1G – Software-based, or Hardware-based

GL MAPS™ HD 40G, 10G, 1G – Rackmount Platforms

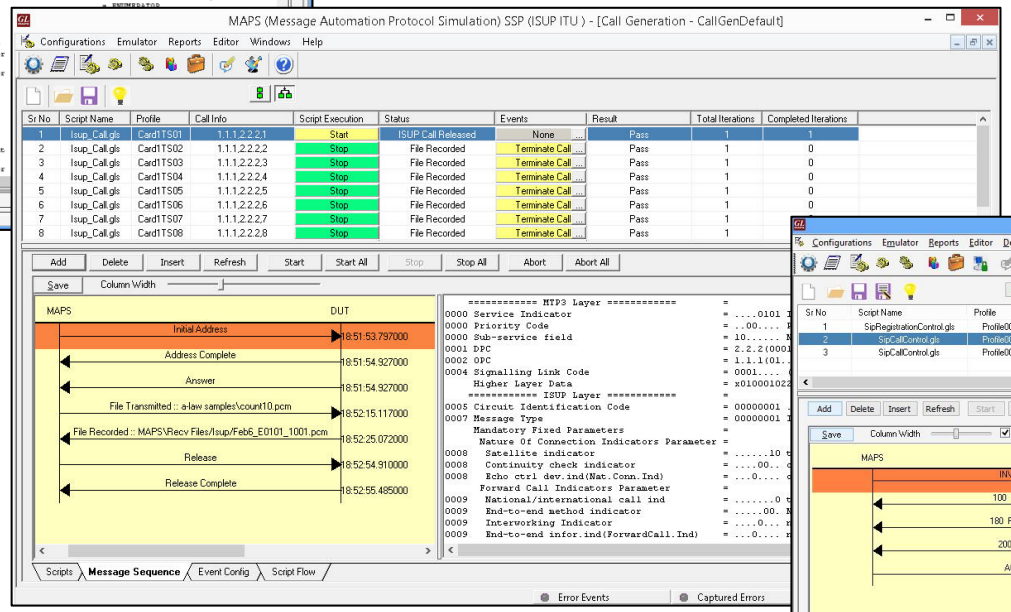
GL Communications

Common Protocol Emulation Framework

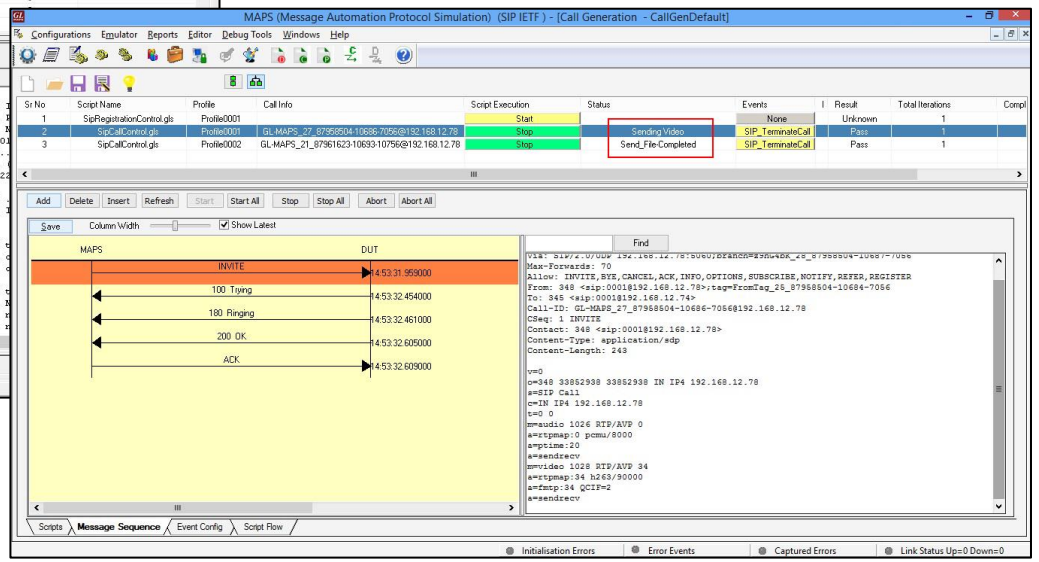
LTE Emulation



SS7 Emulation

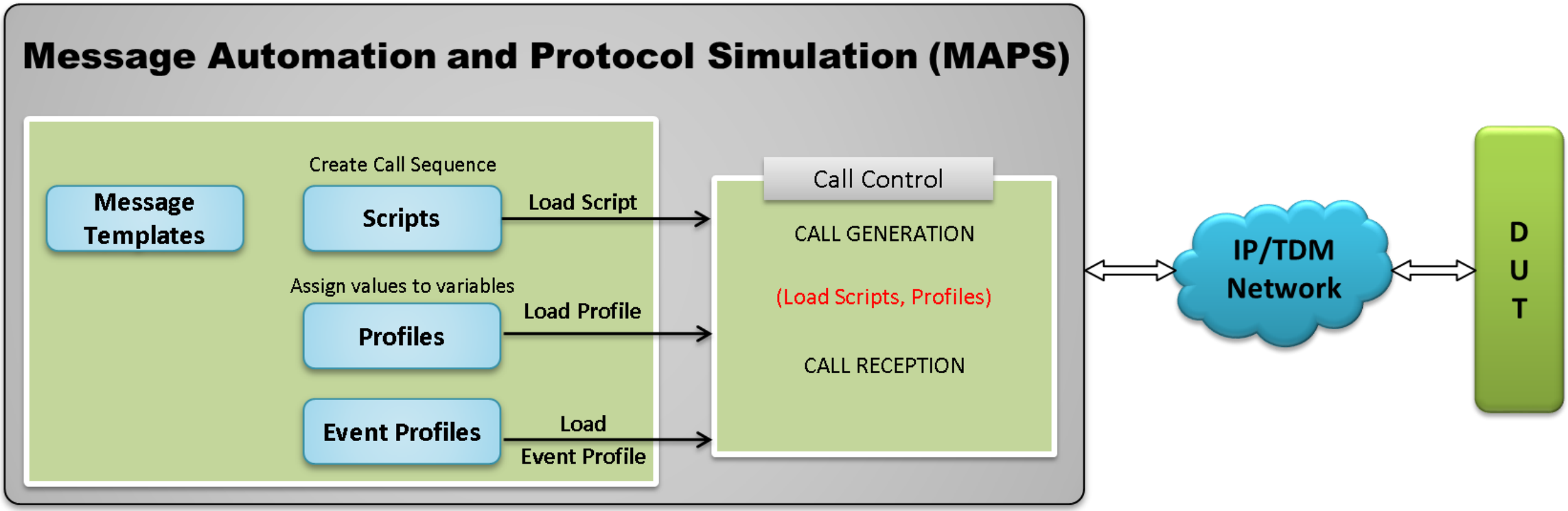


SIP Emulation



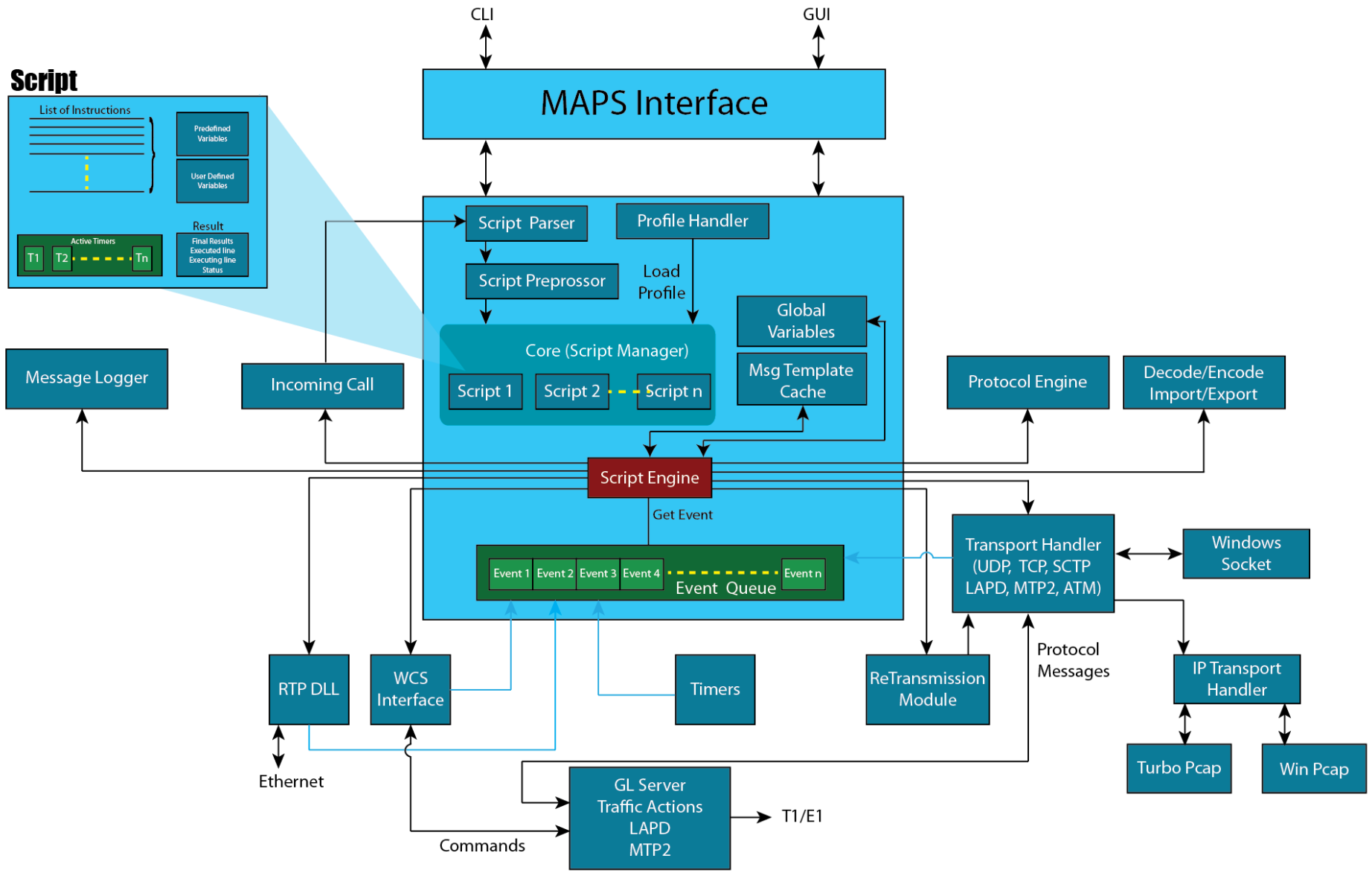
GL Communications

Working Principle



GL Communications

MAPS™ Architecture



GL Communications

Customize Test Scenarios using Scripts

- Unlimited access in creating test scenarios
- Build valid or invalid and conformance test cases
- A simple, easy to learn but very powerful scripting language
- Can be an Extremely simple scripts to test a particular scenario. But Flexible enough to emulate a complete protocol state machine
- A GUI based 'Script Editor' helps to build scripts even before syntax and semantics of the scripting language is familiar

GL Communications

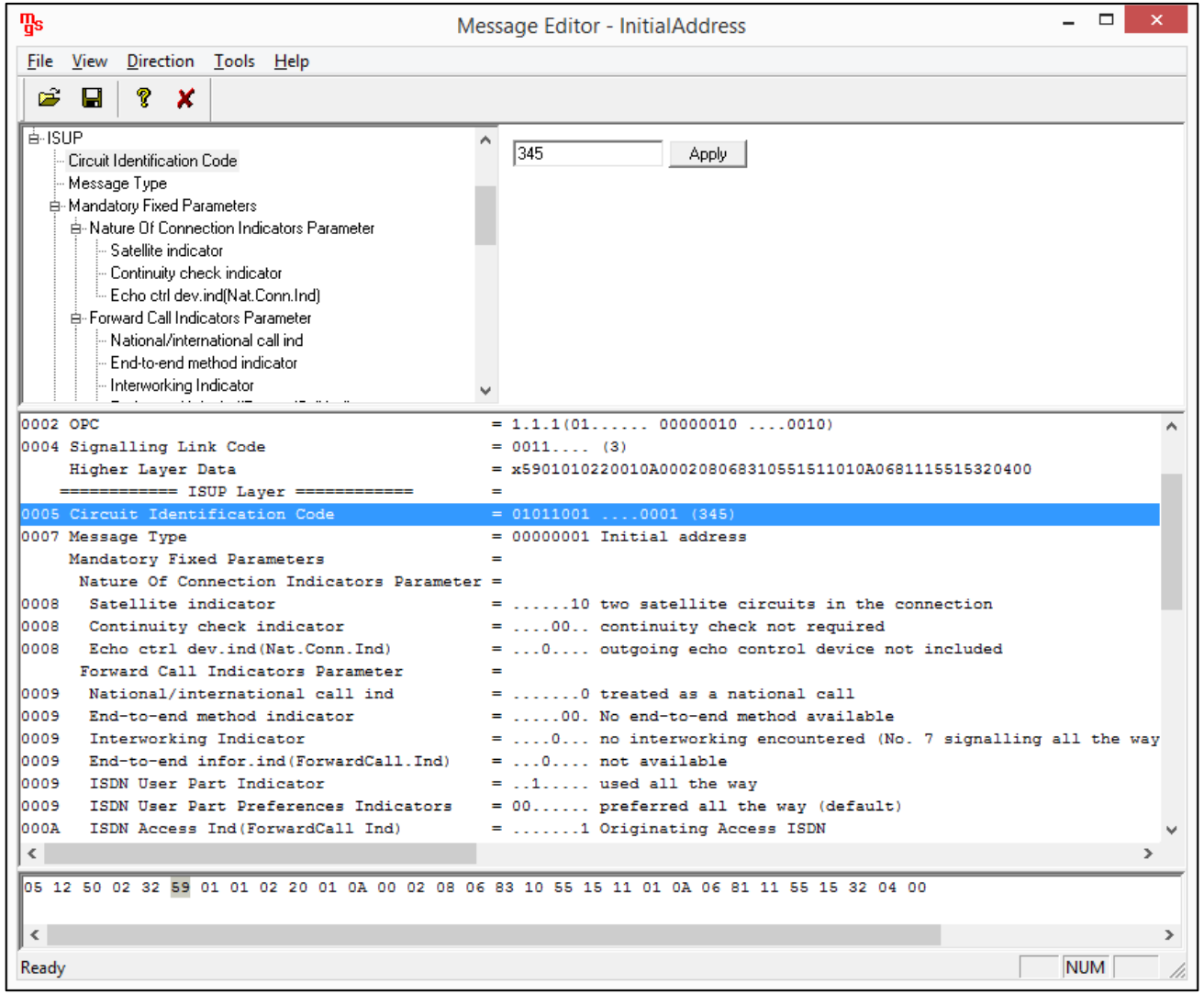
Sample Script

```
Send "Initial Address" "InitialAddressImport";

Recv "Address Complete" "AddressCompleteExport;

Recv "Answer" "AnswerImport";

 TxRx:tx _TDM file: filename = "Vijay.pcm";

Send "Release" "ReleaseImport";

Recv "Release Complete" "ReleaseCompleteExport";
```

GL Communications

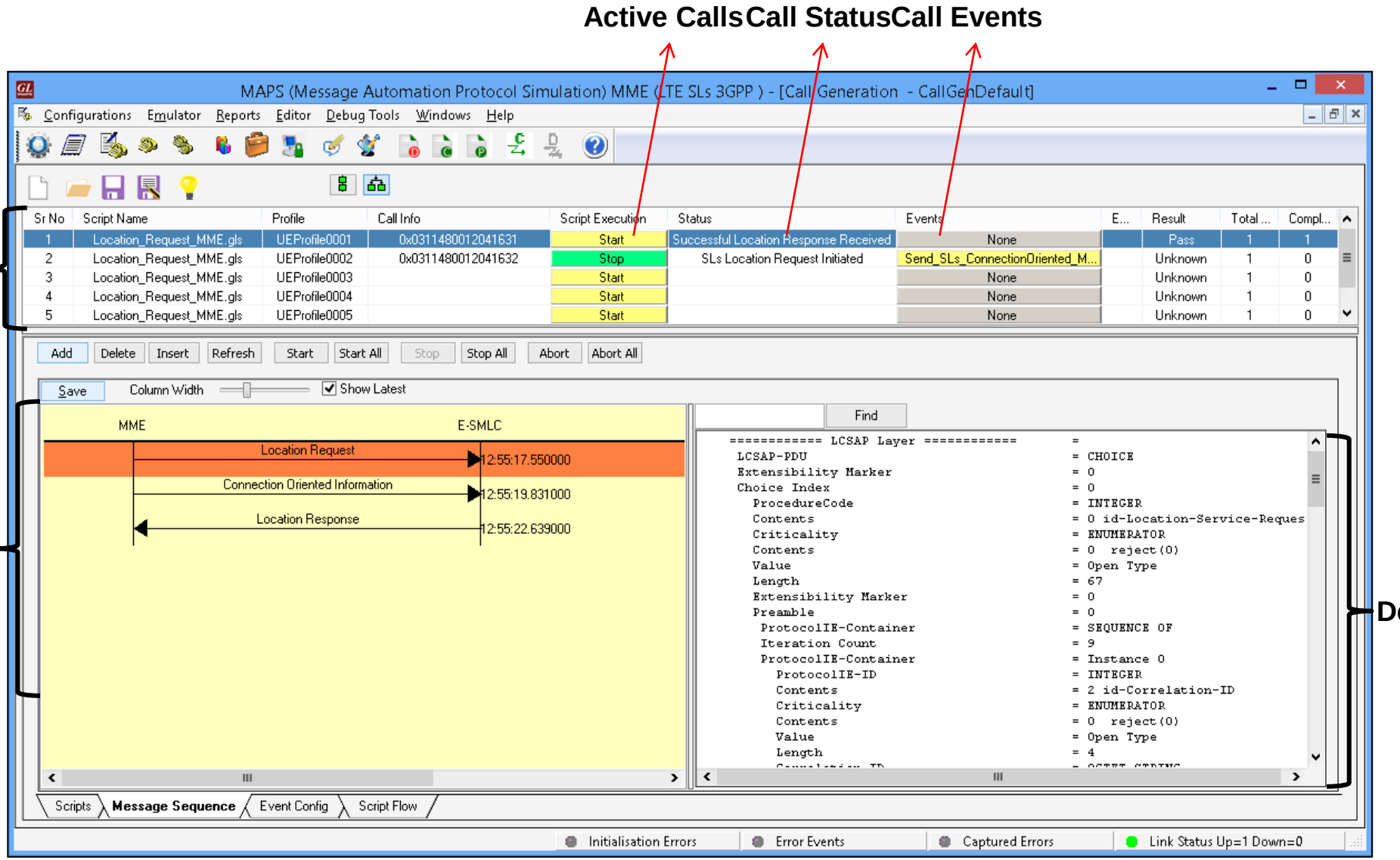
Customize Protocol Messages

Message Editor



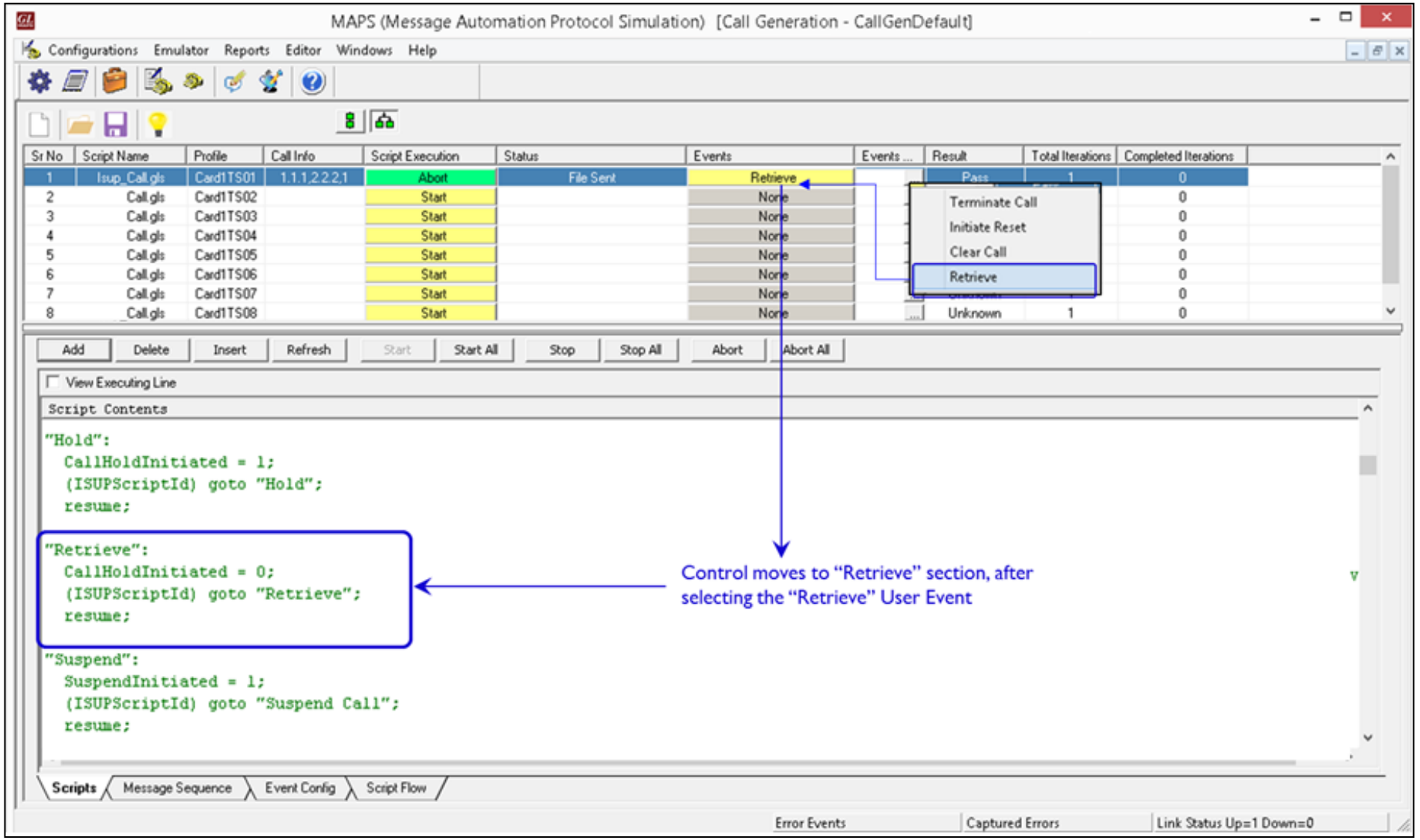
GL Communications

Call Generation



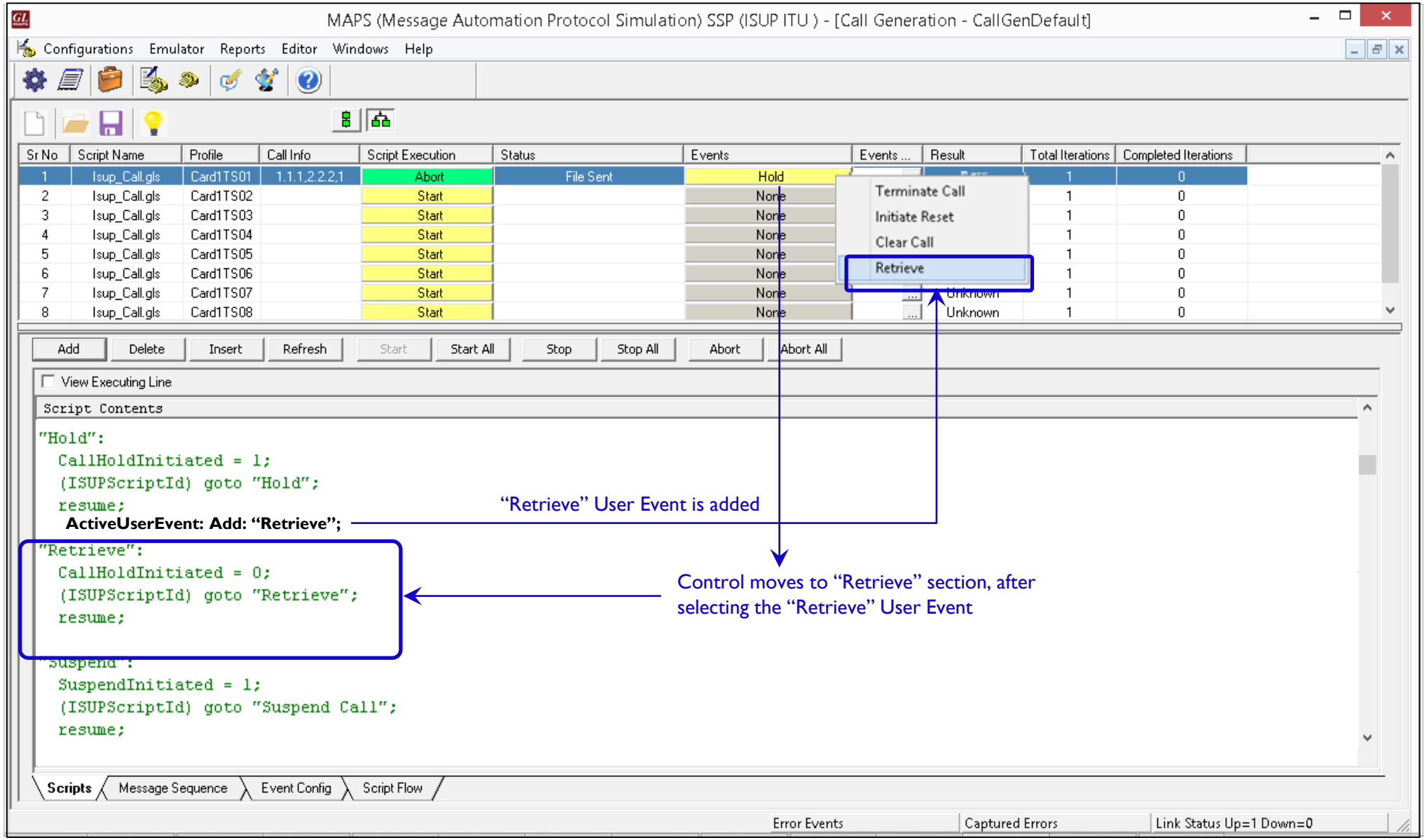
GL Communications

Fine Control over Call Behavior



GL Communications

User Events



GL Communications

Script Content View

Script Contents

```
"PlaceCall"(opc,dpc,cic):
    StartChildScript (ISUPScriptId,"ISUP","ISUP.gls",LoadedProfileName);
    ActiveUserEvent:Add:"Terminate Call","Initiate Reset","Clear Call";
  Status = "ISUP Call Initiated";
  ISUPState = "ISUP CALL INITIATED";
    (ISUPScriptId)goto"ISUPMakeCall":cic,opc,dpc,SLS,NetInd,ConnectionId,StreamID,
    return;

"OnISUPCallInitiated"(opc,dpc,cic):
    ReportEvent (InitialAddress = "Initial Address");
    resume;

"OnISUPCallProgressReceived":
  ReportEvent (AddressComplete = "Address Complete");
    resume;

"OnISUPCallConnected":
    Result="Pass";
  ReportEvent (Answer = "Answer");
  Status = "ISUP Call Connected";
  ISUPState = "ISUP CALL CONNECTED";
  if (StopAll==1)
    goto "Terminate Call":Cause;
    endif
  ActiveUserEvent:Add:"Hold","Suspend";
  ActiveUserEvent:Remove:"Accept Call","Reject Call";
  if (CallDuration != 0)
    starttimer CallDurationTimer CallDuration msec;
  else
    goto "Terminate Call":Cause;
```

Scripts | Message Sequence | Event Config | Script Flow

GL Communications

Script Flow

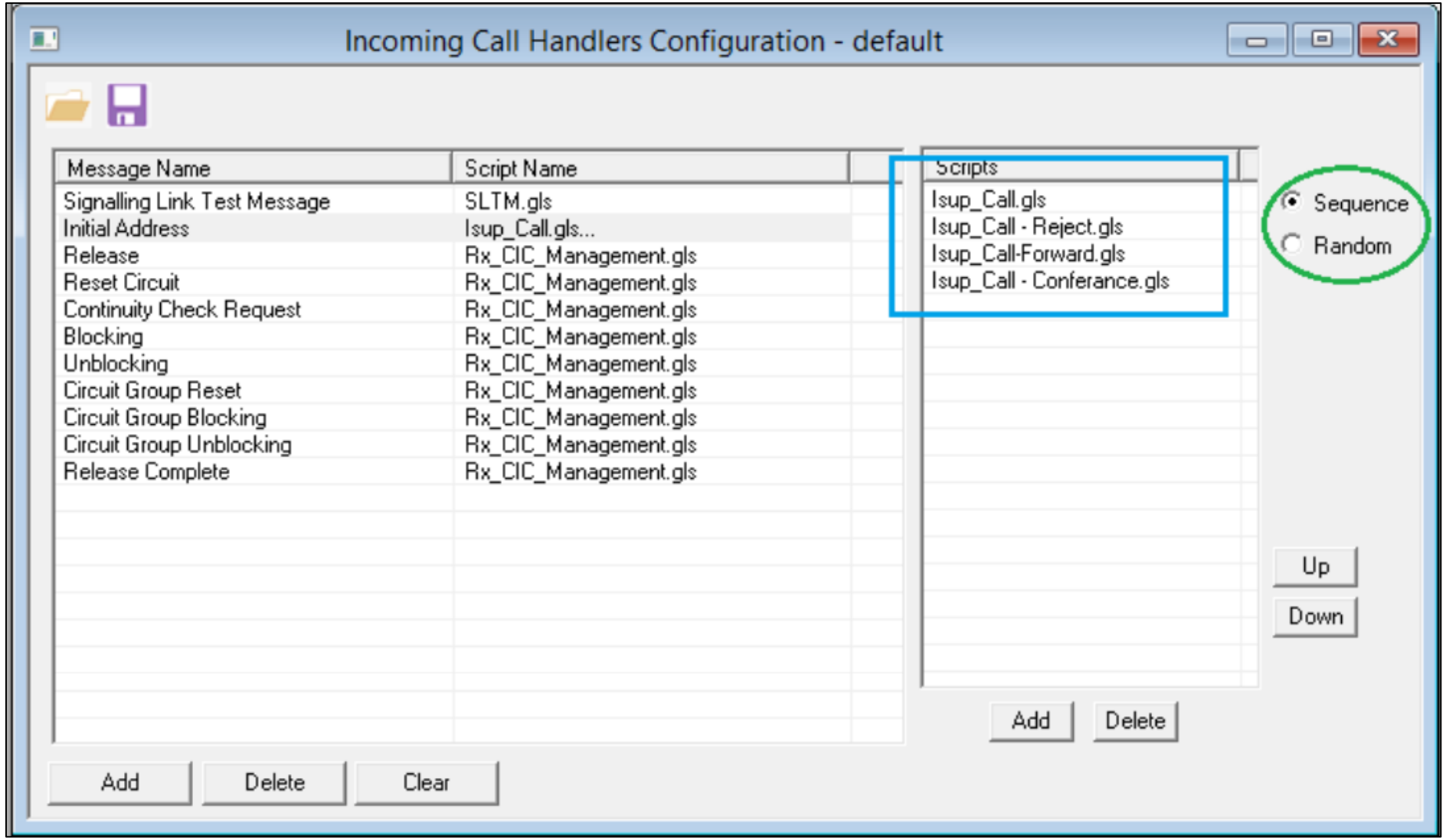
Show Script ID | Show Script Name | Show Subscript Name | Line No | Search | Save

Script ID	Script Name	Subscript Name	Line No	Script Flow
*	Isup_Call.gls		68	goto "PlaceCall":opc,dpc,cic;
*	Isup_Call.gls		94	"PlaceCall"(opc,dpc,cic):
*	Isup_Call.gls		95	StartChildScript (ISUPScriptId,"ISUP","ISUP.gls",LoadedProfileName);
ISUP	ISUP.gls		6	"Init":
ISUP	ISUP.gls		7	ISUPState = "IDLE";
ISUP	ISUP.gls		8	ISUPResult = "Unknown";
ISUP	ISUP.gls		9	SetScriptVariable(ParentScriptId,ISUPResult = ISUPResult);
ISUP	ISUP.gls		10	ParentScriptId = "*";
ISUP	ISUP.gls		11	Cause = 16;
ISUP	ISUP.gls		12	COTExpected = 0;
ISUP	ISUP.gls		13	AddressCompleteSent=0;
ISUP	ISUP.gls		14	KeyIdentifier: opc , dpc, cic ;
ISUP	ISUP.gls		15	ReleaseInitiated = 0;
ISUP	ISUP.gls		16	ReleaseReceived = 0;
ISUP	ISUP.gls		17	CallActive = 0;
ISUP	ISUP.gls		18	MsgHandler : "ISUPMsgHandler";
ISUP	ISUP.gls		19	ReleaseGuardTimerStarted=0;
ISUP	ISUP.gls		21	wait;
*	Isup_Call.gls		96	ActiveUserEvent:Add:"Terminate Call","Initiate Reset","Clear Call";
*	Isup_Call.gls		97	Status = "ISUP Call Initiated";
*	Isup_Call.gls		98	ISUPState = "ISUP CALL INITIATED";
*	Isup_Call.gls		99	(ISUPScriptId)goto"ISUPMakeCall":cic,opc,dpc,SLS,NetInd,ConnectionId,StreamID, ...
ISUP	ISUP.gls		32	"ISUPMakeCall"(cic,opc,dpc,SLS,NetInd,ConnectionId,StreamID, CallingNumber,CalledNu...
ISUP	ISUP.gls		33	send "InitialAddress" "InitialAddressImport" "StreamId" = StreamID "ConnectionI...
ISUP	ISUP.gls		34	if (ContinuityCheckIndicator!=0)
ISUP	ISUP.gls		36	endif
ISUP	ISUP.gls		37	ISUPState="CALL INITIATED" ;
ISUP	ISUP.gls		38	Status = "Call Initiated";
ISUP	ISUP.gls		39	EventLog ("Call Initiated");
ISUP	ISUP.gls		40	starttimer T7 _T7TimeOut;
ISUP	ISUP.gls		41	(ParentScriptId) goto "OnISUPCallInitiated":opc,dpc,cic;

Scripts | Message Sequence | Event Config | **Script Flow**

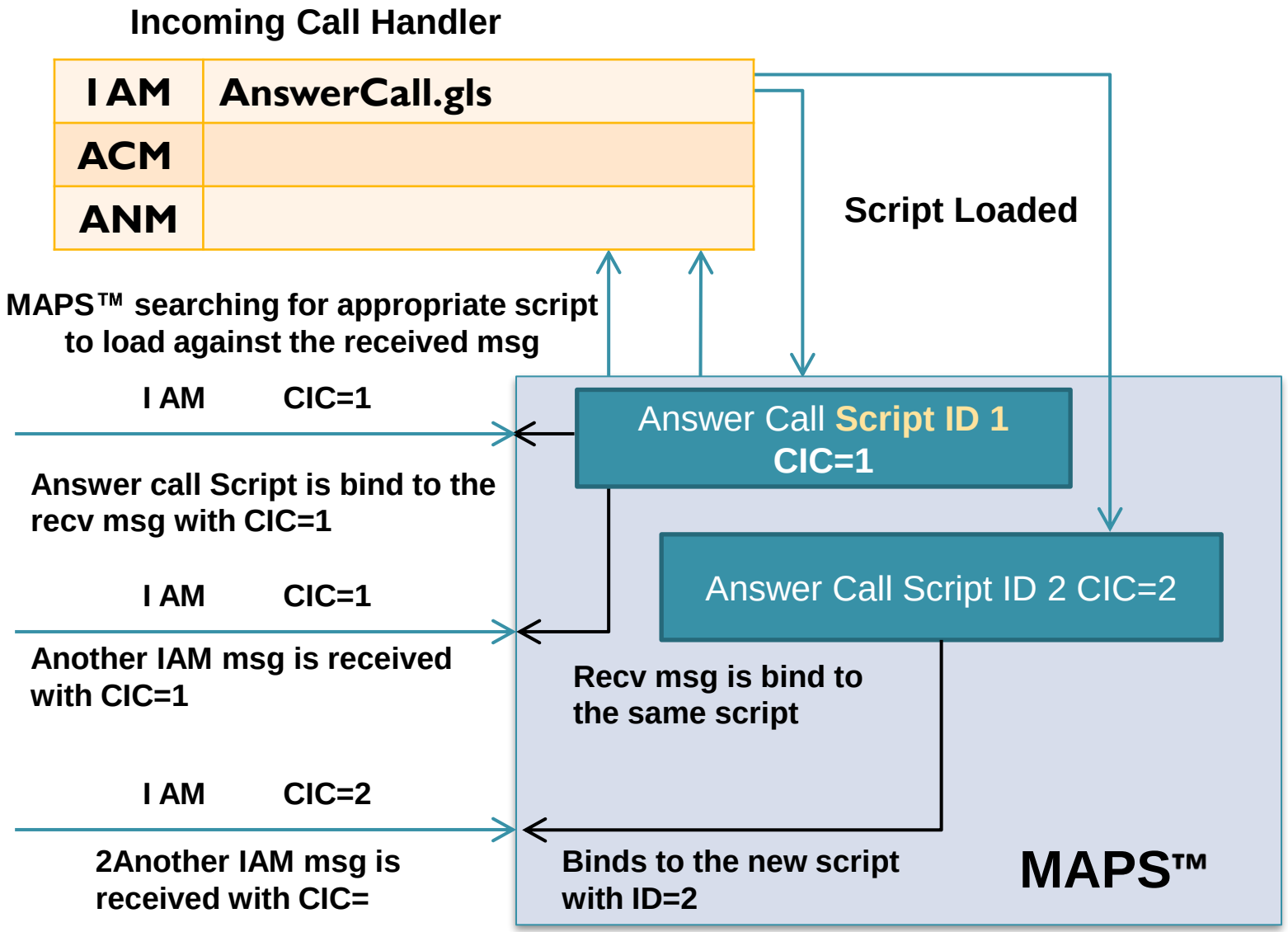
GL Communications

Incoming Call Handler



GL Communications

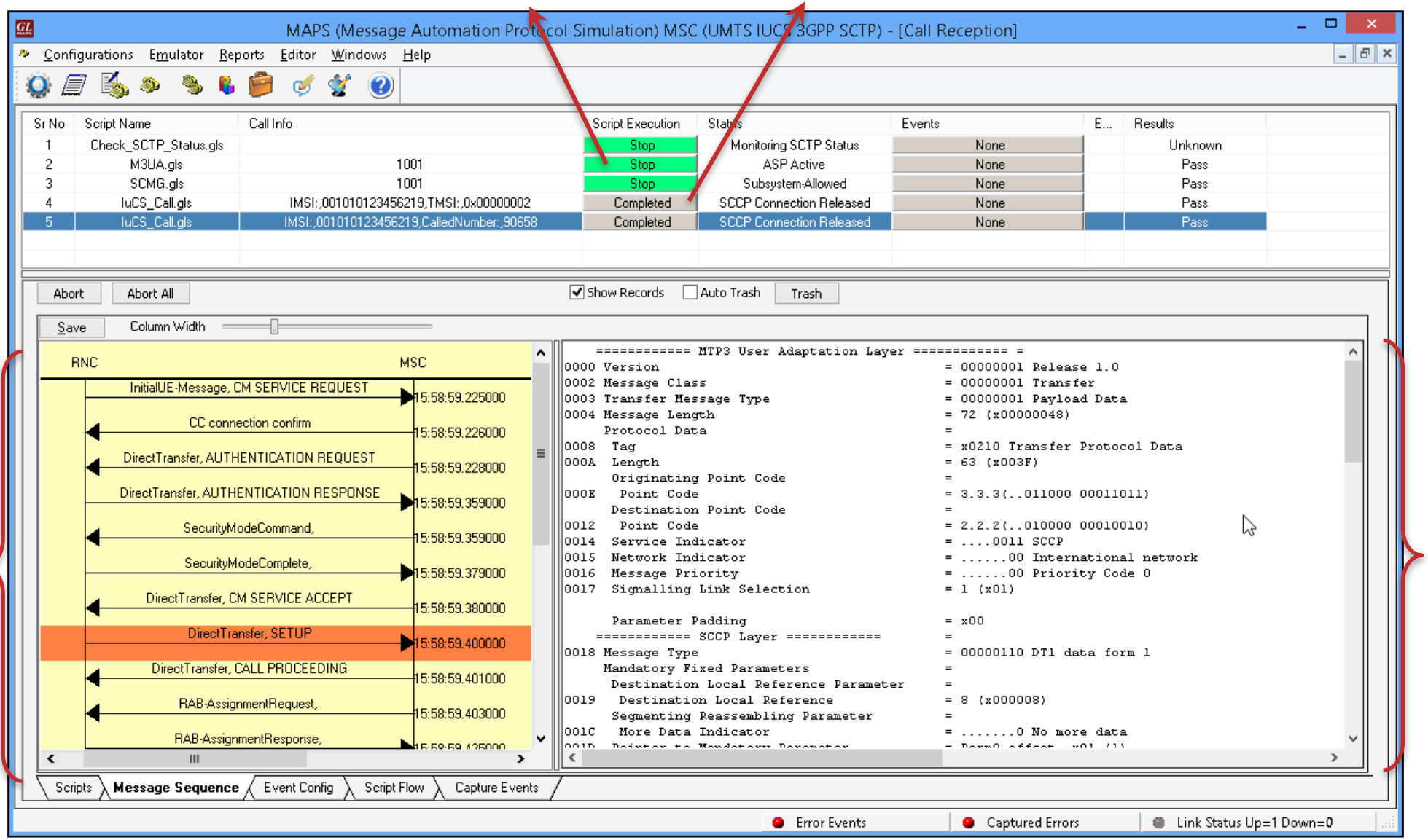
Incoming Call Handler (Contd.)



Call Reception

Active Calls

Completed Calls



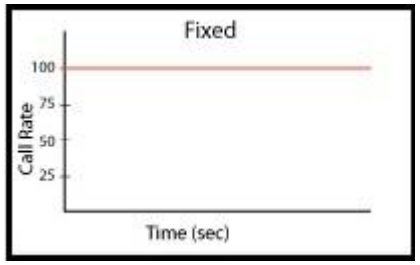
Message Sequence

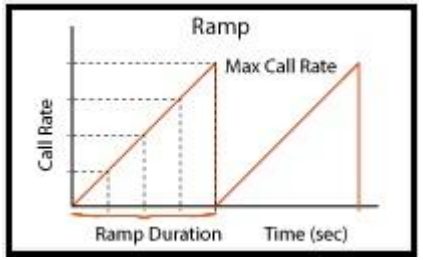
Decoded Message Details

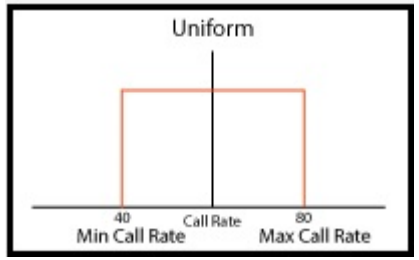
GL Communications

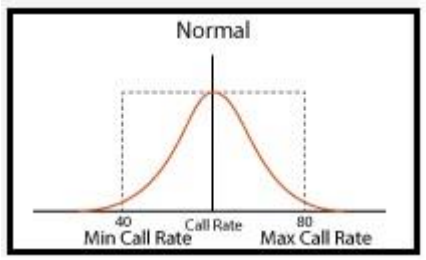
Load Generation

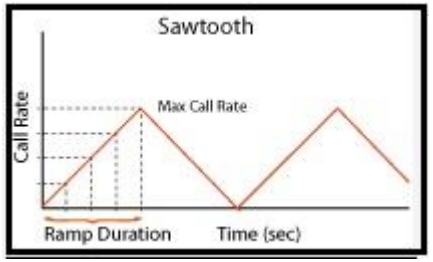
- Stability/Stress and Performance testing using Load Generation
- Different types of Load patterns to distribute load
- User can load multiple patterns for selected script
- User configurable Test Duration, CPS, Maximum and Minimum Call Rate etc.

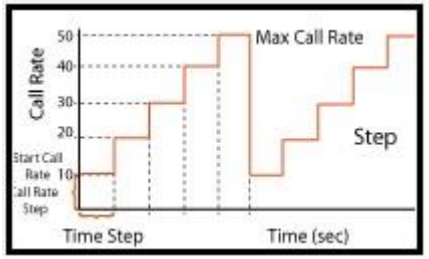


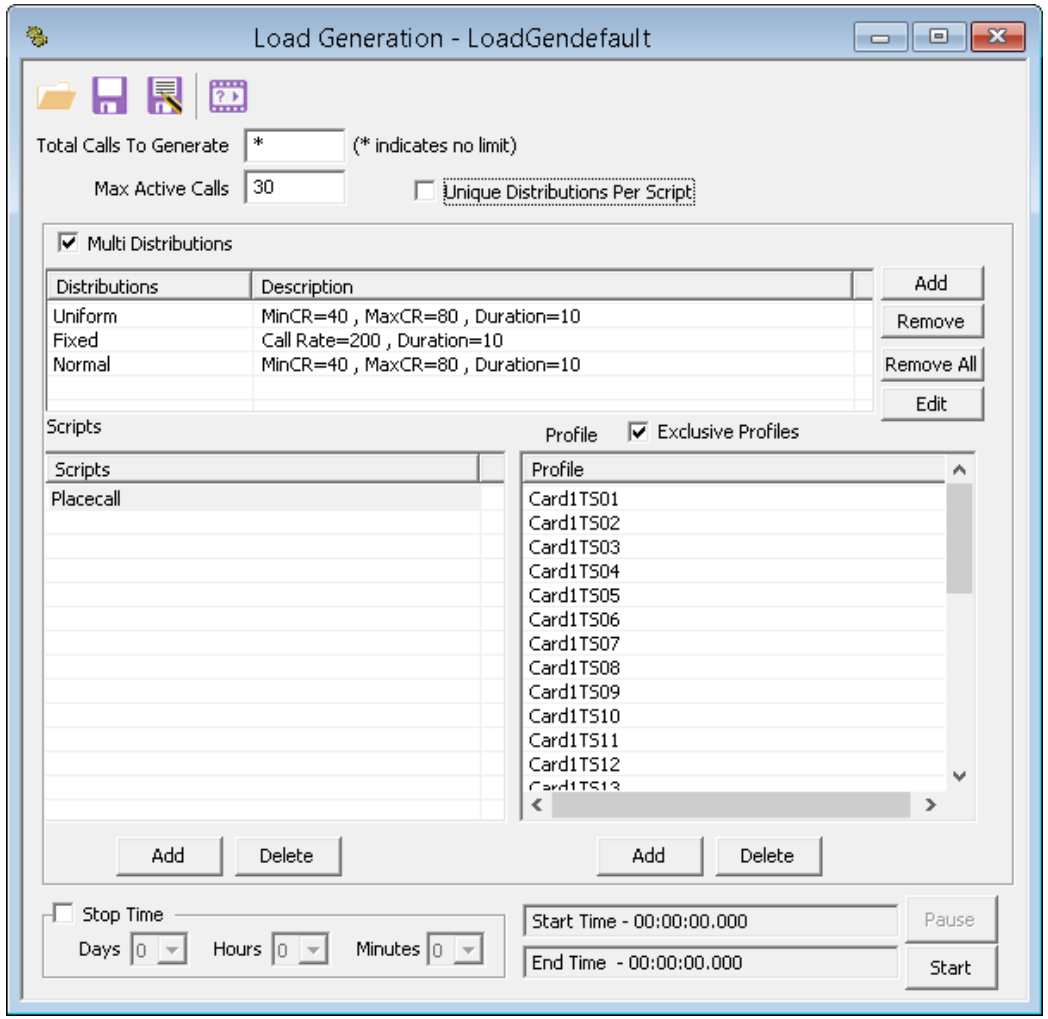










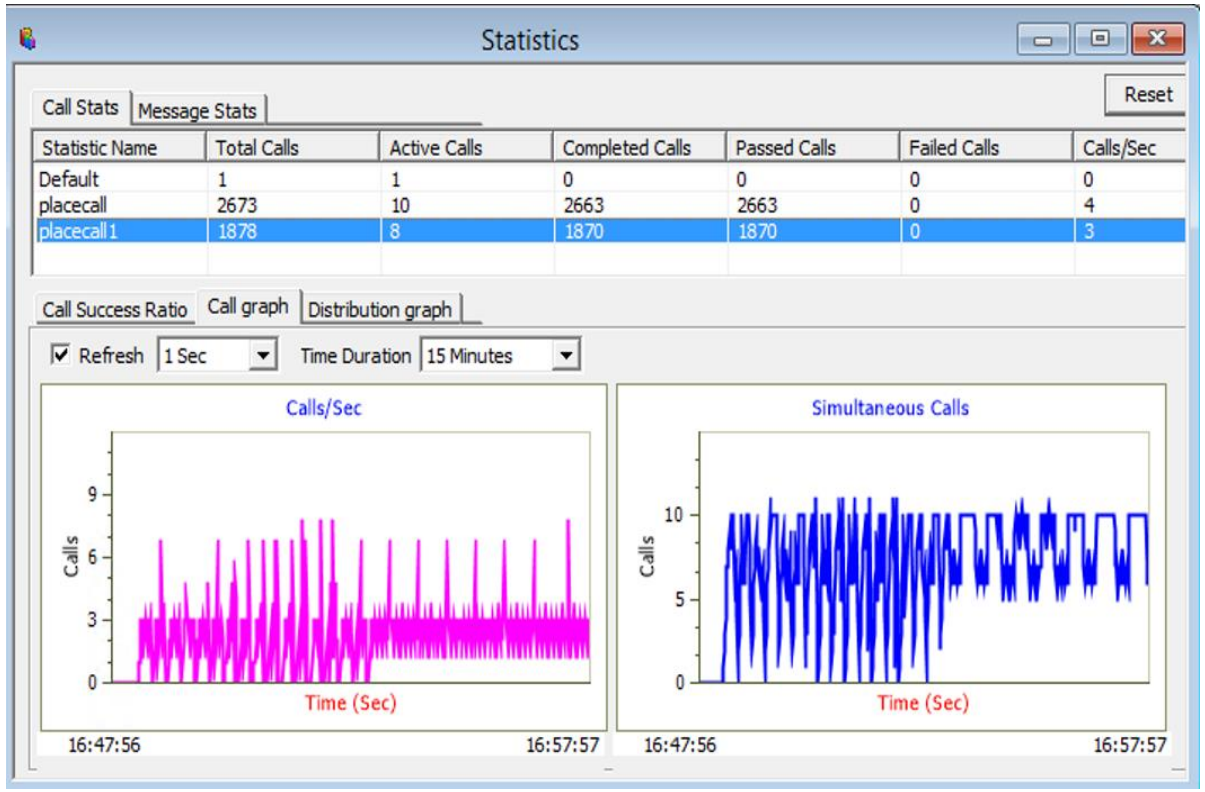


GL Communications

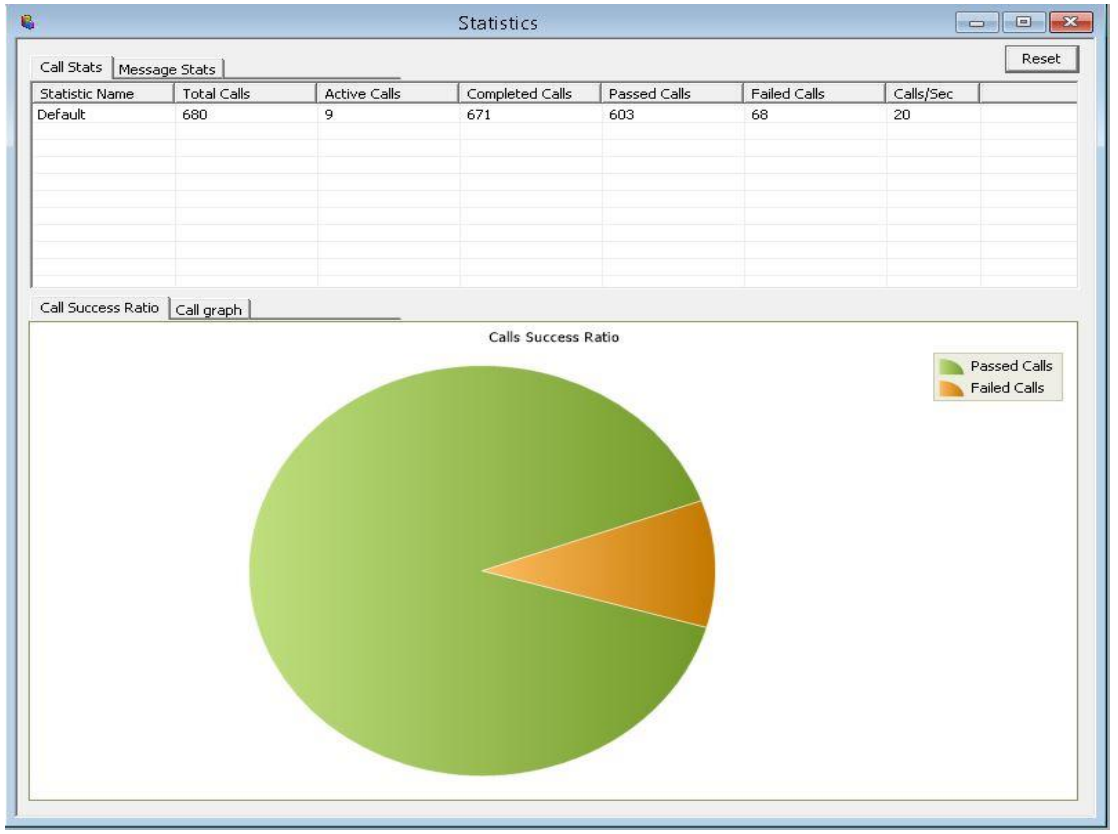
Success Call Ratio Statistics

MAPS™ Features

Call Graph



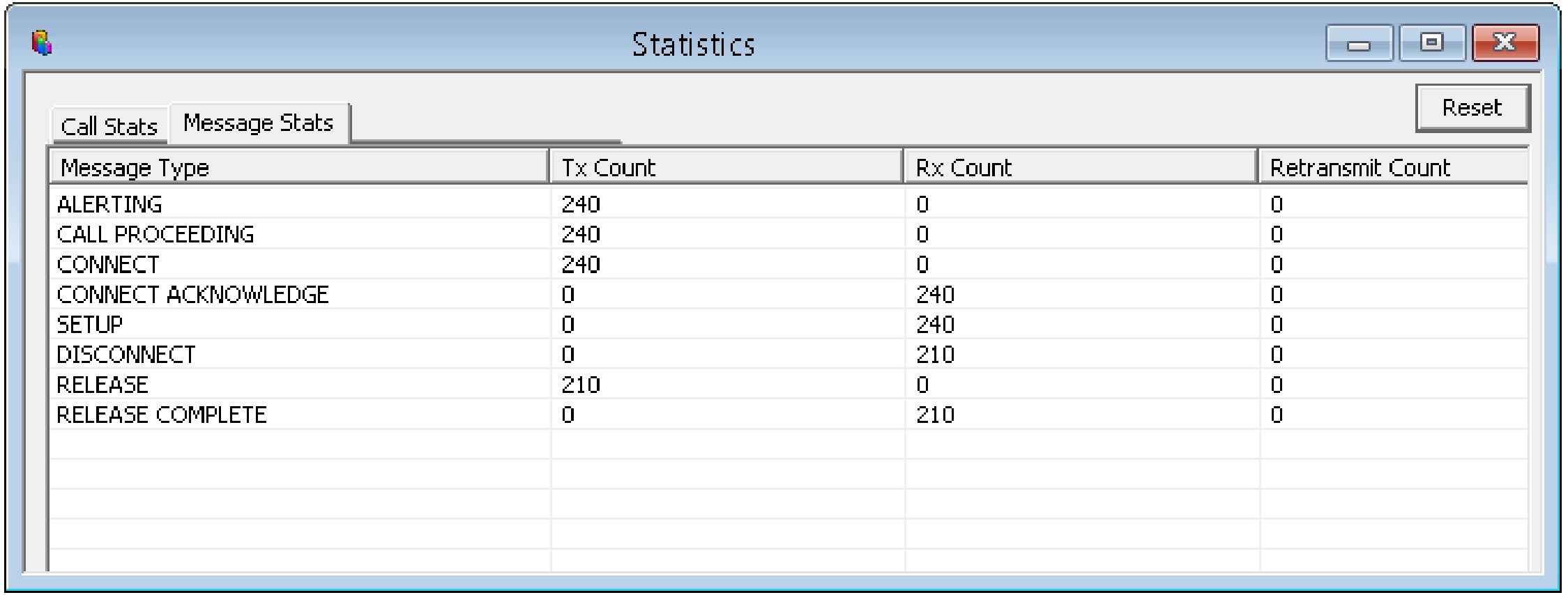
Call Stats



GL Communications

Message Statistics

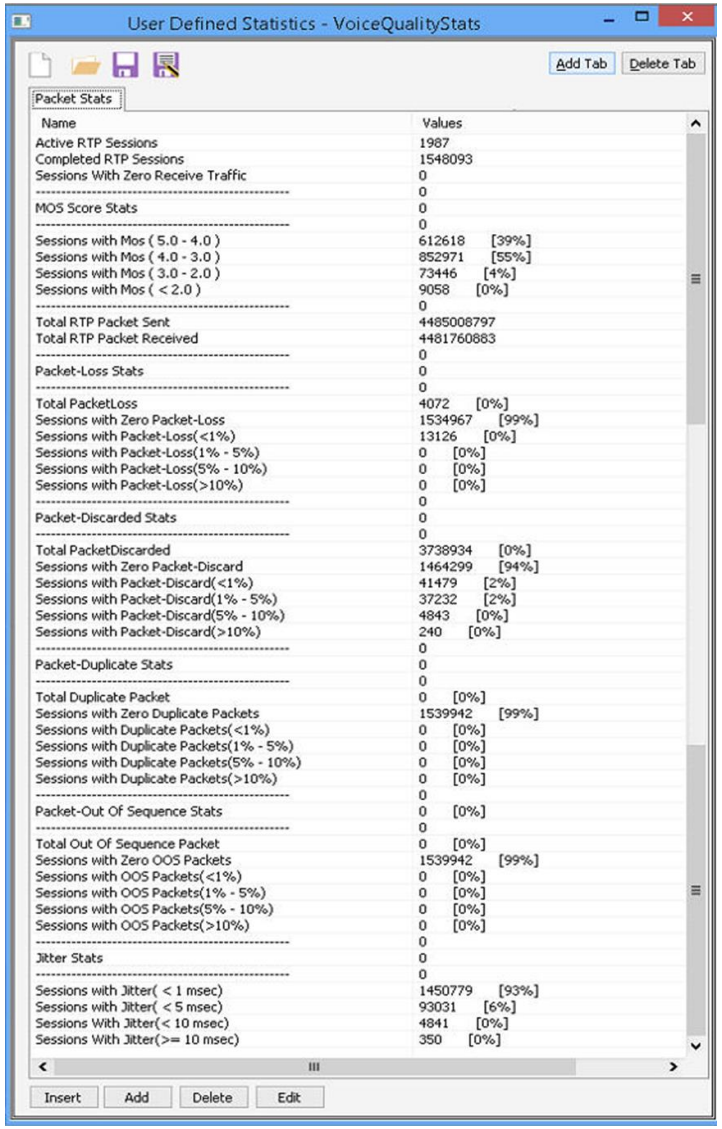
- Message Stats provides a running tabular log of all messages transmitted, retransmitted and received during the session
- Provides an easy way to monitor the reception of error responses during load generation

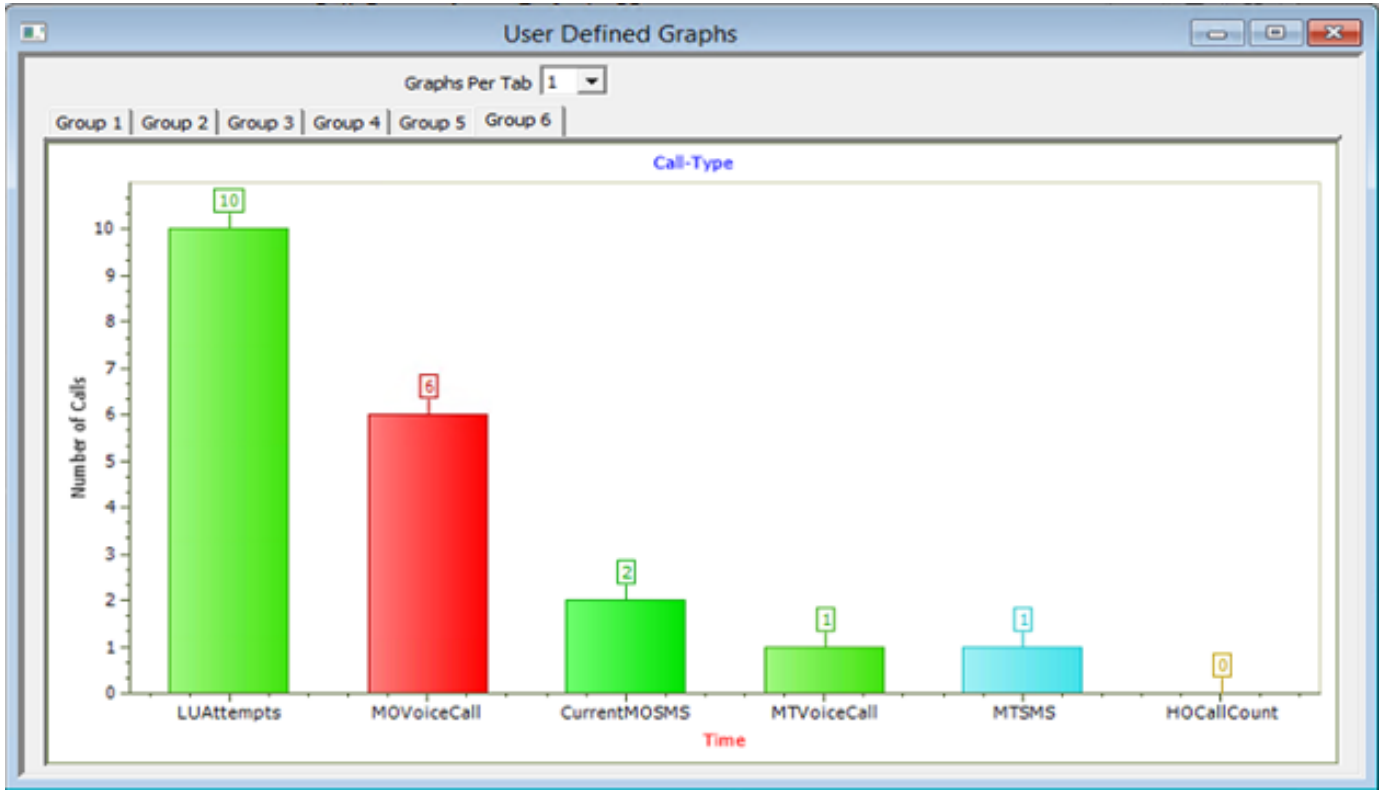


Message Type	Tx Count	Rx Count	Retransmit Count
ALERTING	240	0	0
CALL PROCEEDING	240	0	0
CONNECT	240	0	0
CONNECT ACKNOWLEDGE	0	240	0
SETUP	0	240	0
DISCONNECT	0	210	0
RELEASE	210	0	0
RELEASE COMPLETE	0	210	0

GL Communications

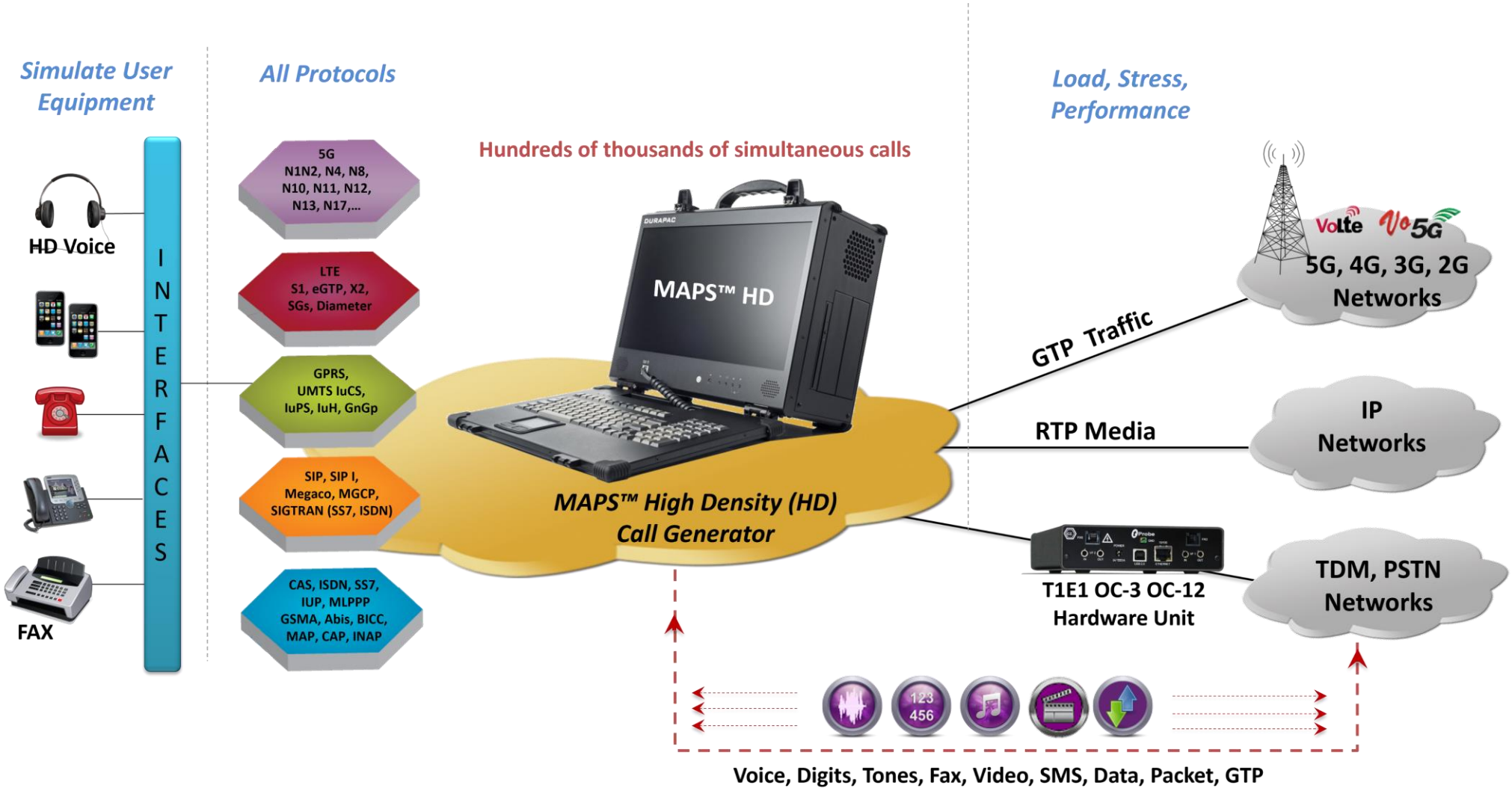
User Defined Graphs and Statistics





GL Communications

Traffic Simulation



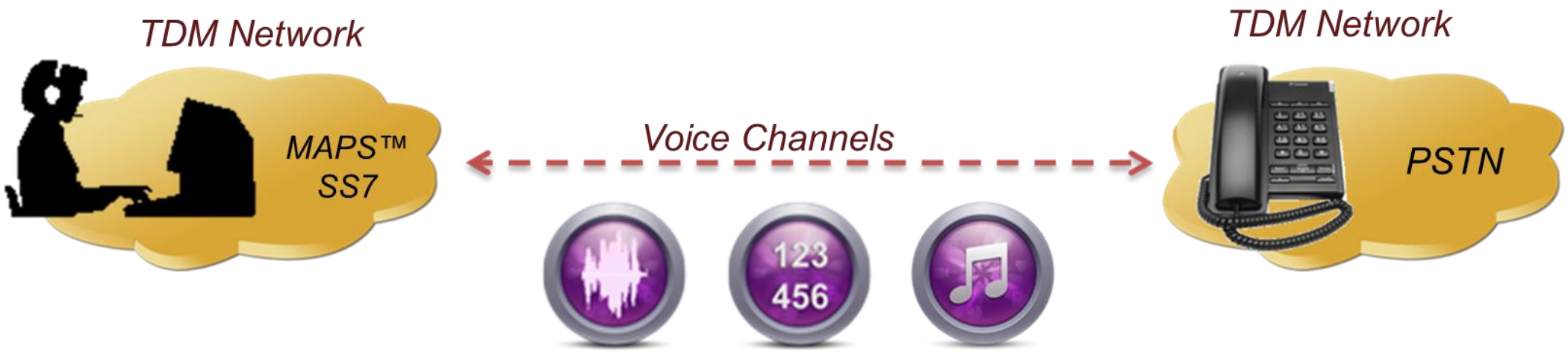
GL Communications

Supported RTP Codecs

- **PCMU/PCMA:** 64kbps, 8000Hz, VAD
- **G.722/G.722.1:** 24/32/64kbps, 16000Hz, No VAD
- **G.729/G.729B:** 8kbps, 8000Hz, VAD
- **GSM 6.10 FR:** 13.2kbps, 8000Hz, No VAD
- **GSM EFR:** 12.2kbps, 8000Hz Yes VAD
- **GSM:** 5.6kbps, 8000Hz, Yes VAD
- **G.726:** 16/24/32/40kbps, 8000Hz, Yes VAD
- **AMR:** 4.75/5.15/5.9/6.7/7.4/7.95/10.2/12.2kbps, 8000Hz, Yes VAD *(OPTIONAL LICENSE)*
- **AMR WB:** 4.75/5.15/5.9/6.7/7.4/7.95/10.2/12.2kbps, 16000Hz, Yes VAD *(OPTIONAL LICENSE)*
- **EVRC:** 1/8, 1/2, 1 rate, 8000Hz, No VAD *(OPTIONAL LICENSE)*
- **EVRC_B:** 1/8, 1/2, 1 rate, 8000Hz, Yes VAD *(OPTIONAL LICENSE)*
- **EVRC_C:** 1/8, 1/2, 1 rate, 16000Hz, Yes VAD *(OPTIONAL LICENSE)*
- **SMV:** Modes 0,1,2 and 3, 8000Hz, No VAD *(OPTIONAL LICENSE)*
- **ILBC:** 15.2/13.33kbps, 8000Hz, No VAD
- **SPEEX:** 8kbps, 8000Hz, Yes VAD
- **SPEEX WB:** 11.2kbps, 16000Hz, Yes VAD

GL Communications

TDM Traffic Simulation



Tx

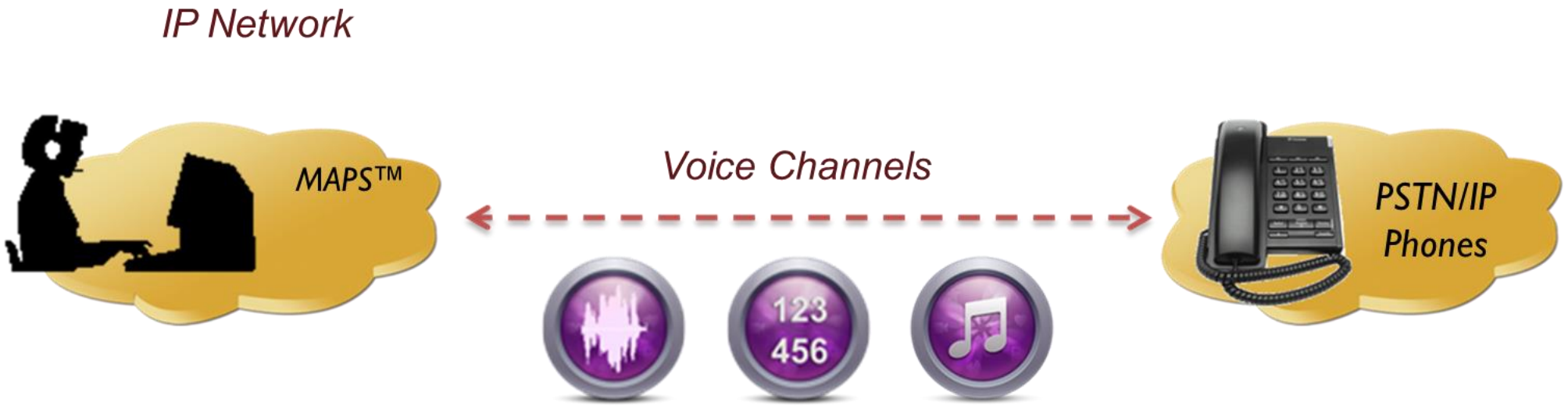
- *Pre recorded PCM files*
- *DTMF, MF, MFR2B and MFR2F Digits*
- *User Defined Tones*
- *FAX*
- *AAL2*

Rx

- *PCM files*
- *DTMF, MF, MFR2B and MFR2F Digits*
- *User Defined Tones*
- *FAX*
- *AAL2*

GL Communications

RTP Traffic Simulation



Tx

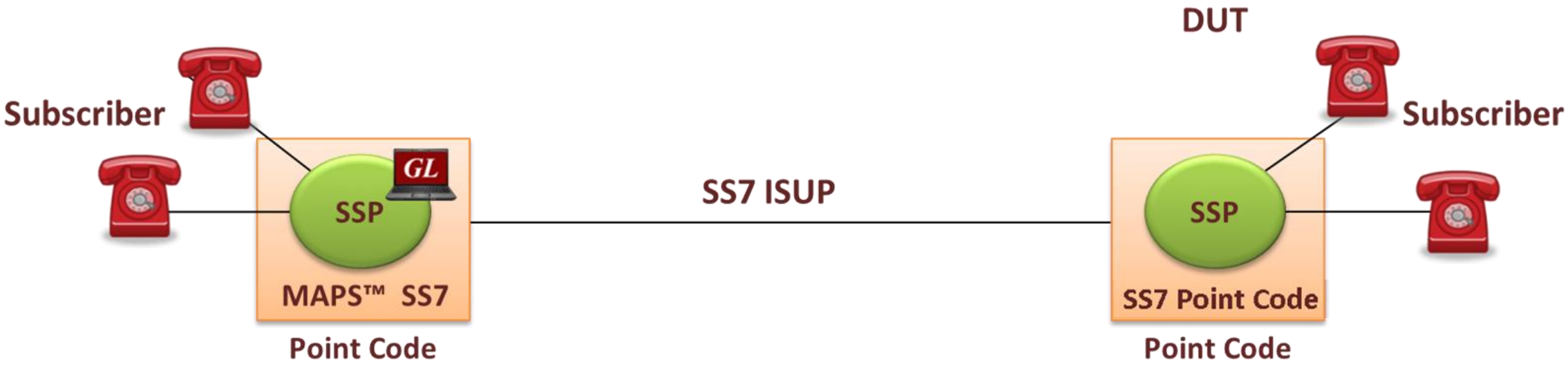
- *Pre recorded GLW files*
- *DTMF, MF Digits*
- *User Defined Tones*
- *Insert Voice*
- *FAX T.30*

Rx

- *GLW files*
- *DTMF, MF Digits*
- *User Defined Tones*
- *FAX T.30*

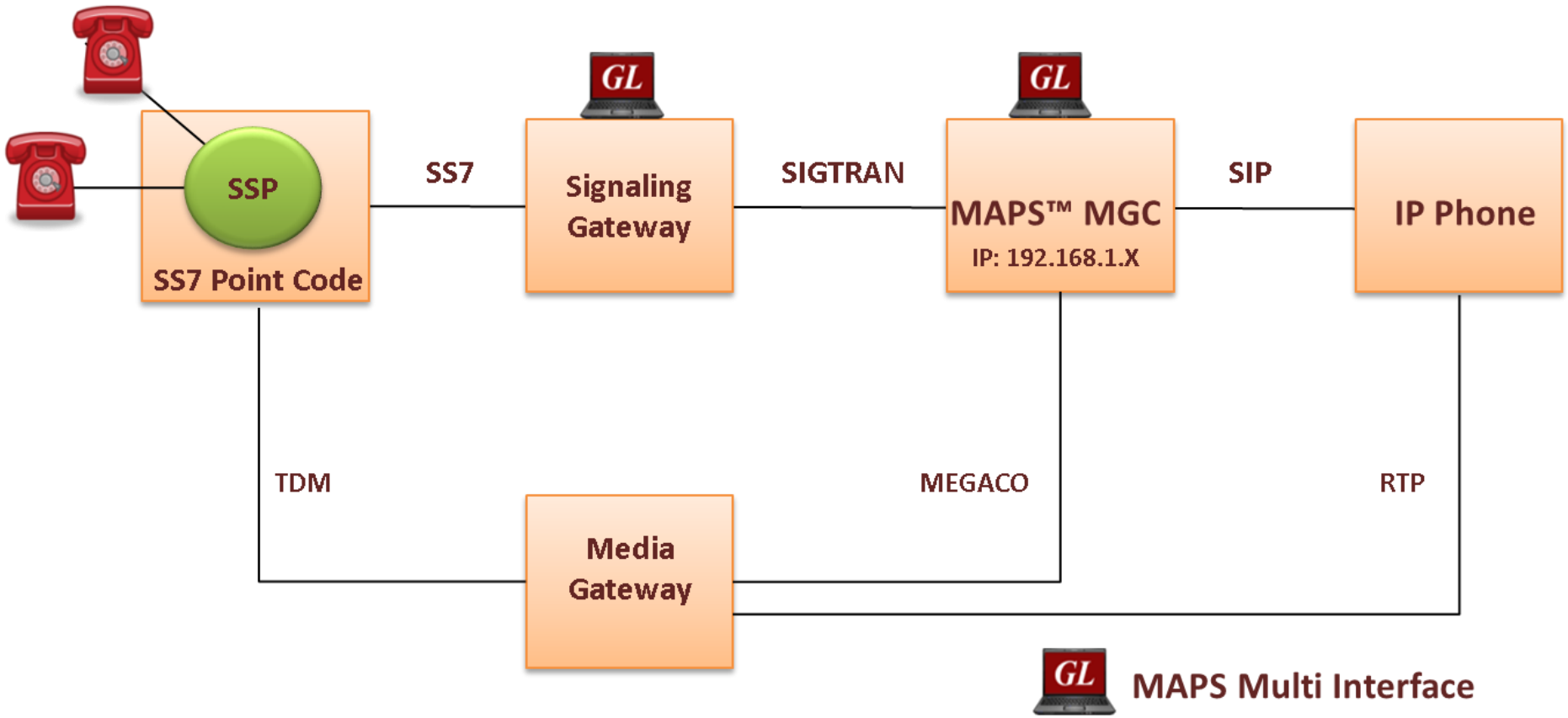
GL Communications

Single Interface Simulation



GL Communications

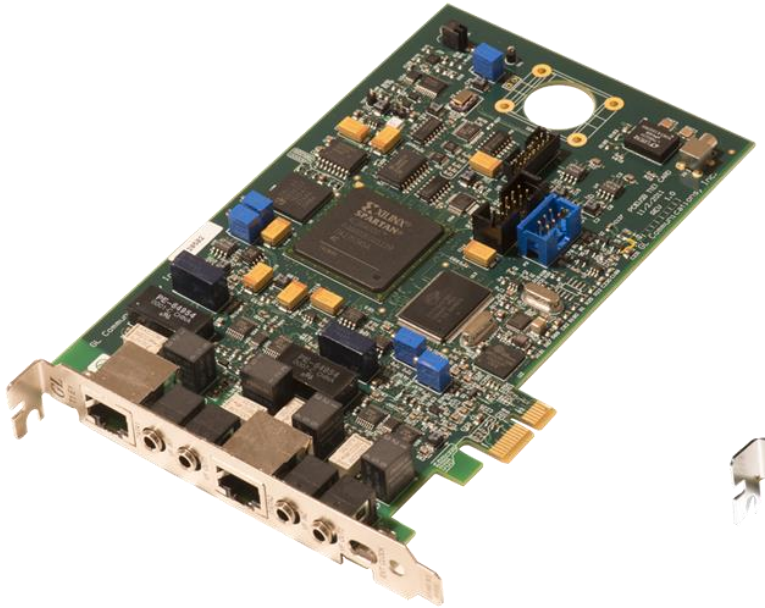
Multi Interface Simulation



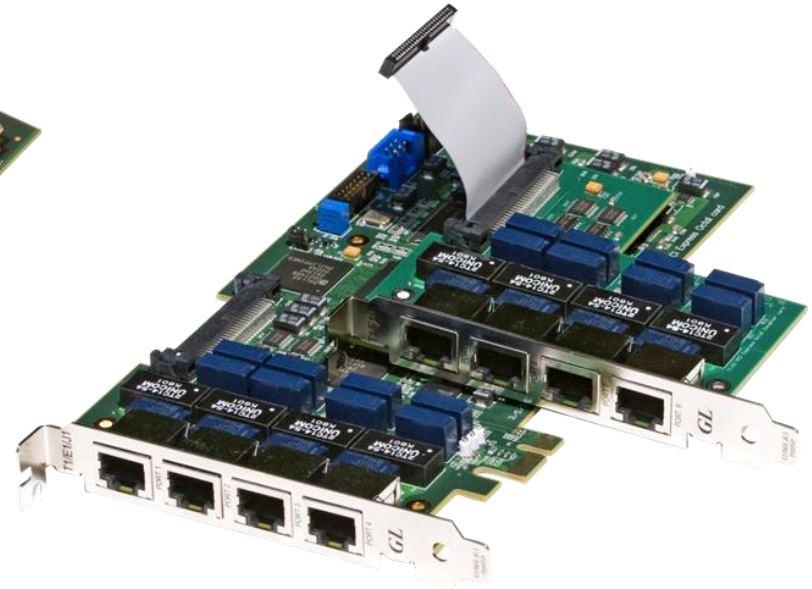
GL Communications

Multiple Transport Support

tProbe™ - Portable USB based T1 E1 VF FXO FXS and Serial Datacom Analyzer



Dual T1 E1 Express (PCIe) Board



Quad / Octal T1 E1 PCIe Card

IP Hardware

- IP variants of MAPS™ can be run on any modern Windows® server
- A typical i7 platform will be able to handle ~2000 concurrent RTP sessions through a conventional server-grade NIC
- We also offer an HD (High Density) appliance which can deliver up to 20,000 concurrent RTP sessions per Unit of rack space

GL Communications

High Density (HD) RTP Traffic Simulation

- Rackmount/lunchbox network appliance with 4x1GigE or 8x1GigE NIC
- Signaling calls can be transmitted over UDP and TCP, IPv4 and IPv6, and TLS for secure transport
- Emulates around 50,000 to 100,000 user endpoints.
- Up to 250 calls per second (with RTP traffic)
- For UDP transport, scales up to 64,000 simultaneous calls for each appliance (i.e. 8,000 RTP media sessions per port) with duplex RTP traffic
- Manage 10+ MAPS™ systems with single point of control from master controller

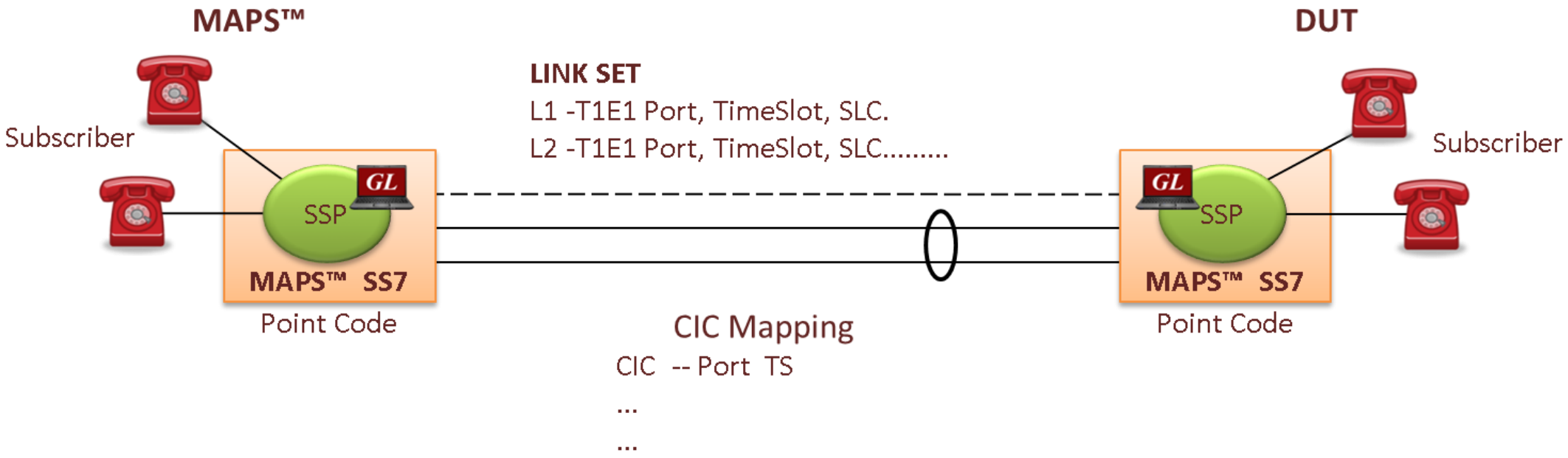
MAPS™ HD Lunchbox Appliance

MAPS™ HD Rackmount Appliance

GL Communications

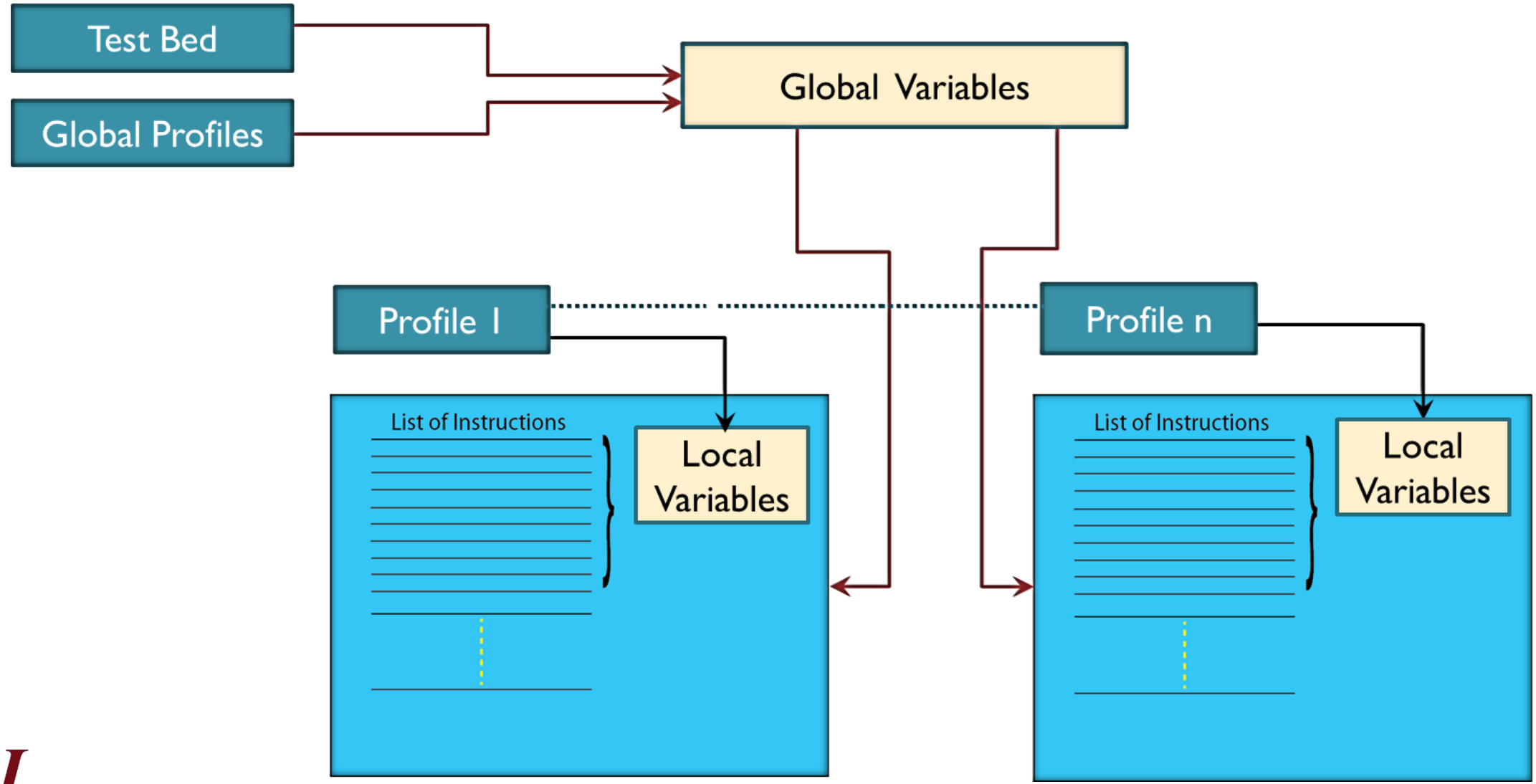
Introduction to MAPS™ Configurations

- Testbed Setup
- Global Configuration
- Profiles



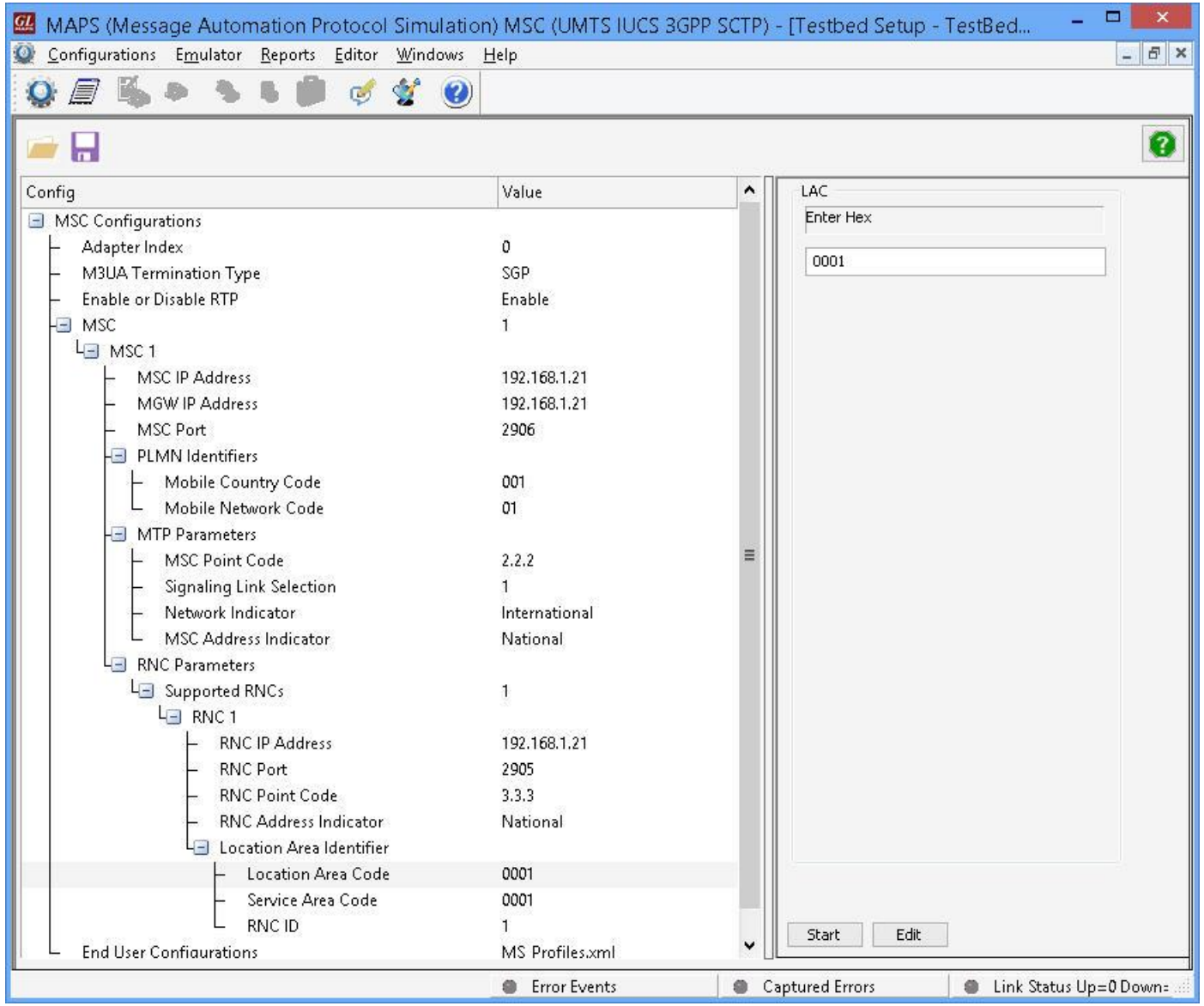
GL Communications

Local and Global Variables



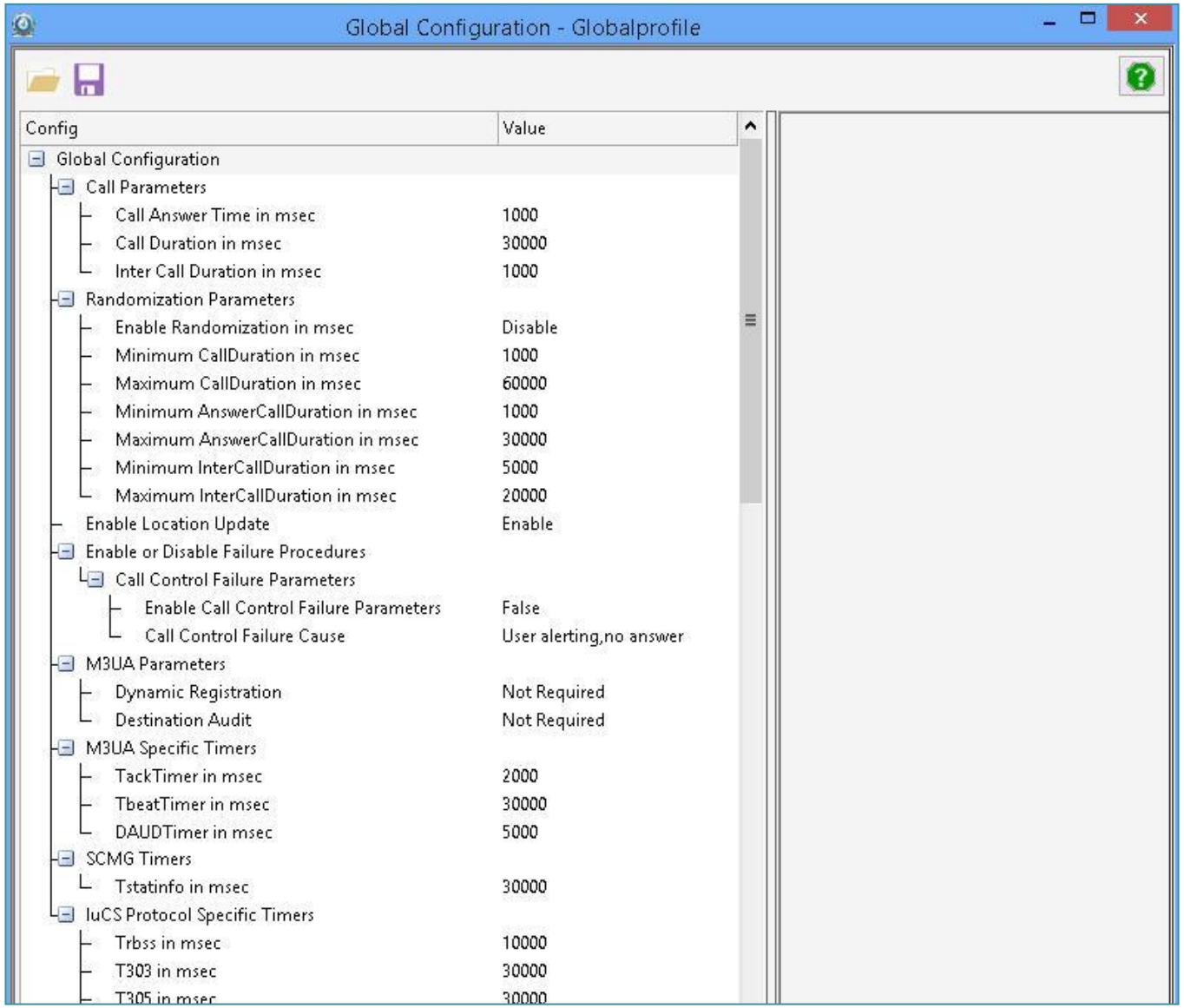
GL Communications

Testbed Configuration



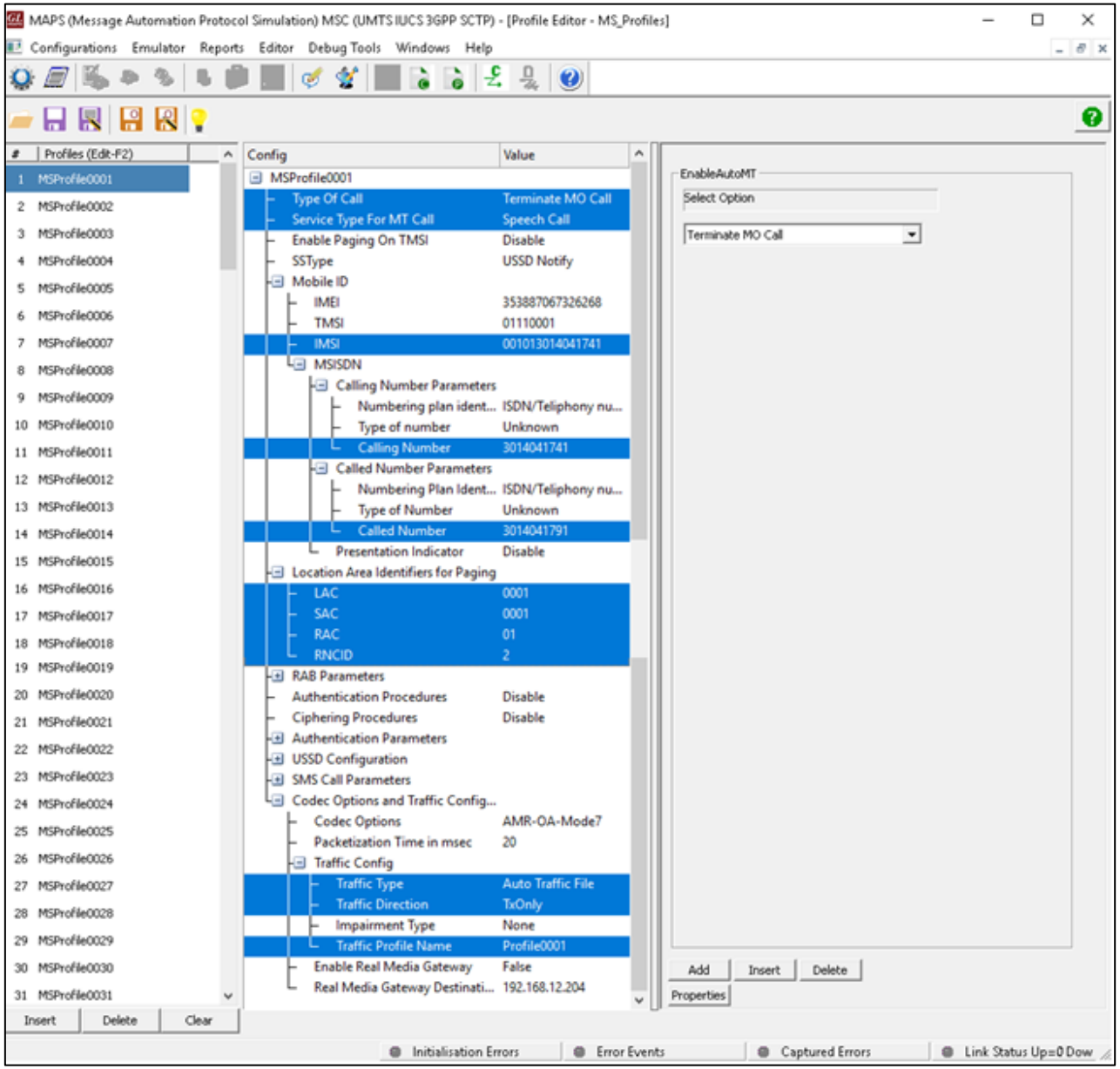
GL Communications

Global Configuration



GL Communications

Sample Profile



GL Communications

MAPS™ Scripting

Script Variants

- MAPS™ Scripts can be written in different ways as we have flexible commands such as **Go to**, **IF Else IF**, **Timers**, **Actions**, **User Events** etc.

- **Two Types of Scripting**
  - Simple, Non-Event driven
  - Event Driven

- **Non-Event driven:** Defines flow sequentially without monitoring any events. These can be small and simple scripts using send and receive actions

- **Event Driven:** Defines flow on basis of user selected events. Using Event Driven scripting one can achieve Protocol State Machines as per protocol specifications

Structure of Non Event Driven Script

```
//Script Description ..................

//Initalization Section
---------
---------

//Action section

Send "MessageName" "ImportFile Name";
Recv "Message" "ExportFile Name";
Result = "Pass";
State = ".............";
Status = ".............";
Exit;
```

GL Communications

Script Events

- **Message Handler:** On Receipt of any Message Event control move to defined section in script

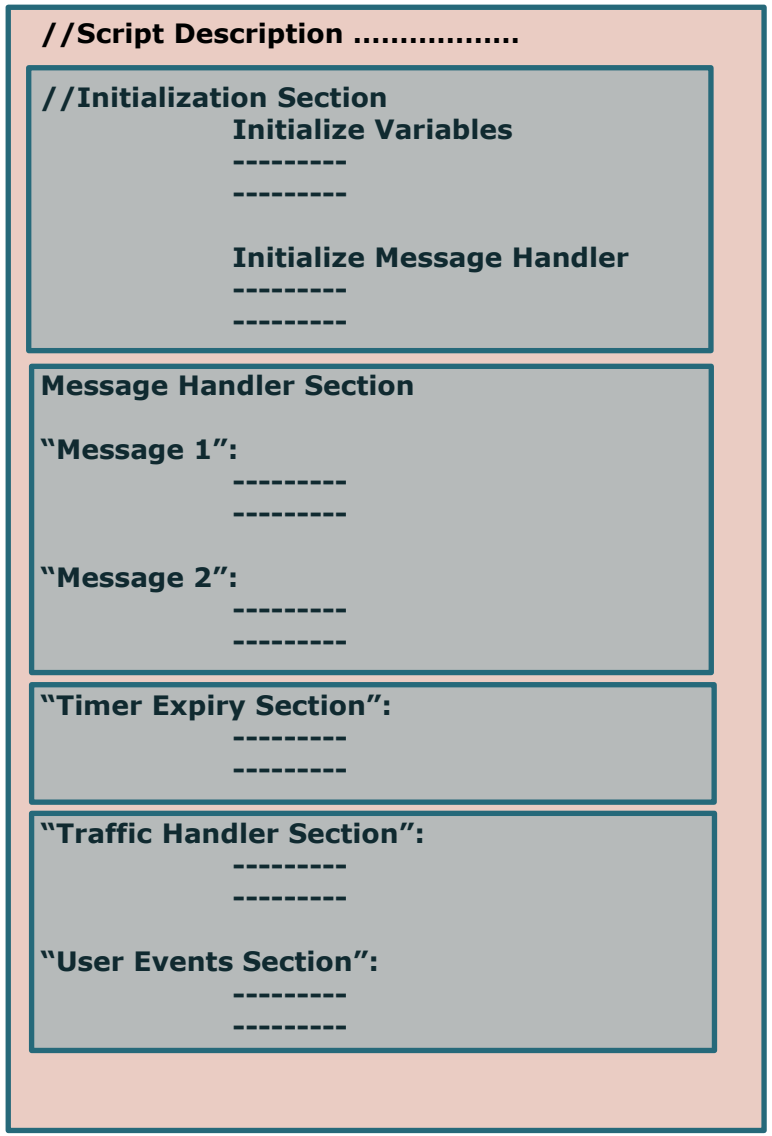
- **Traffic Event:** On detection of any traffic actions, Control move to detected Traffic Event Section like “Digits Detected”, Tone Detected”, etc.

- **Timer:** On Expiry of Timer, control moves to respective Timer Expiry section

- **User Events:**
  - Within scripts: Goto “Label”
  - User Intervention: User Event
  - Intervention from another Script: Apply Event to another script

GL Communications

Structure of Event Driven Script



GL Communications

Scripts

Below call flow scenario using MAPS Script

Send “Initial Address” “InitialAddressImport”;

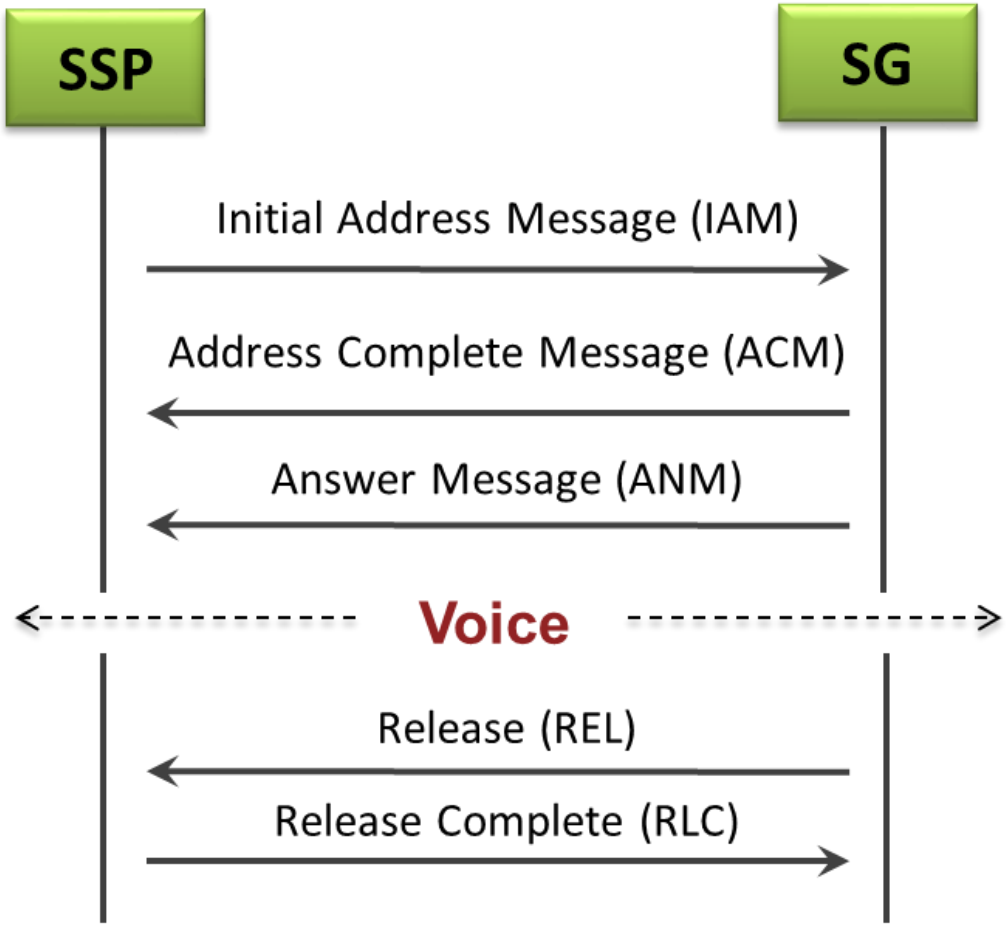
Recv “Address Complete” “AddressCompleteExport;

Recv “Answer” “AnswerImport”;

TxRx:tx _TDM file: filename = “Vijay.pcm”;

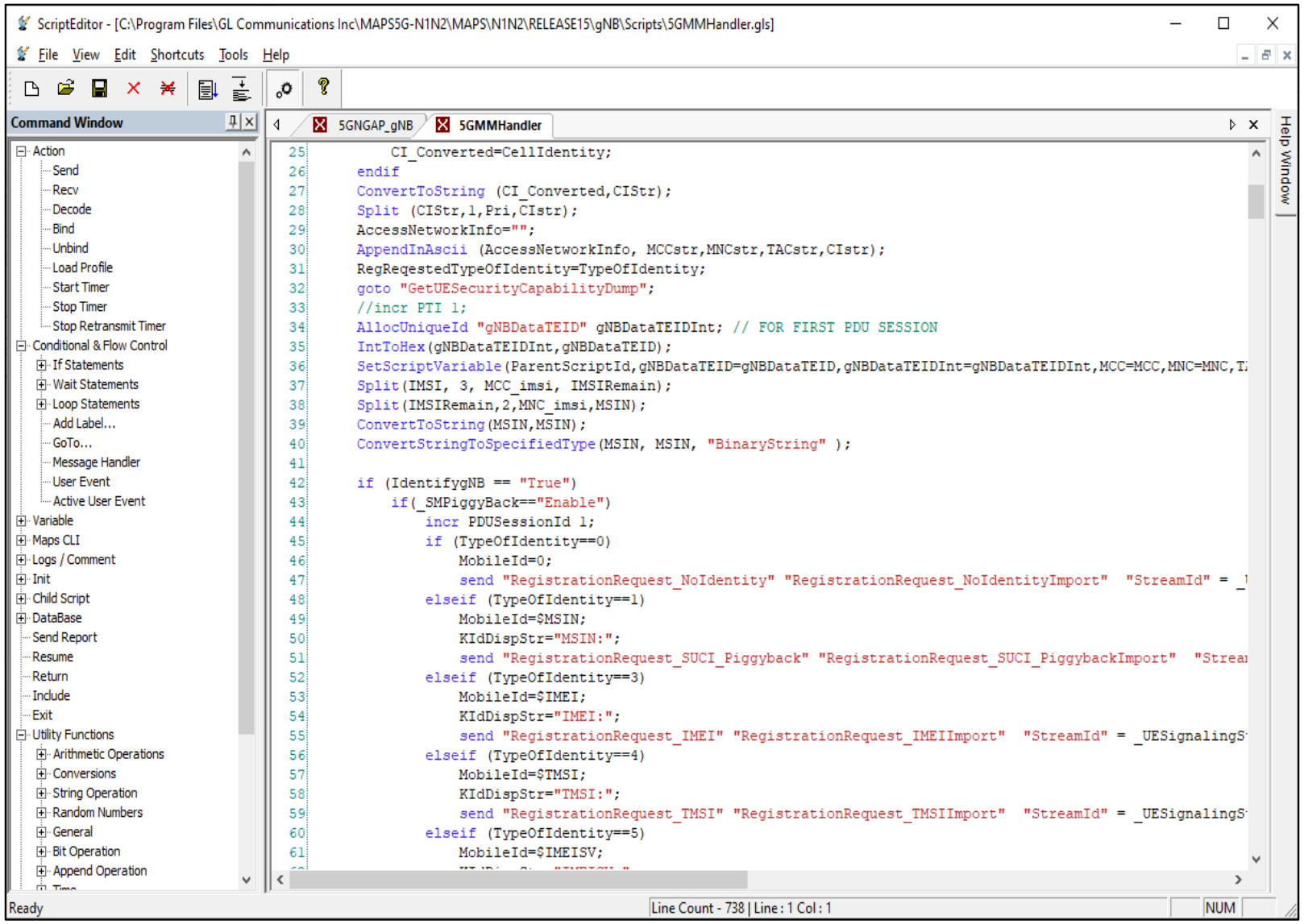
Send “Release” “ReleaseImport”;

Recv “Release Complete” “ReleaseCompleteExport”;



GL Communications

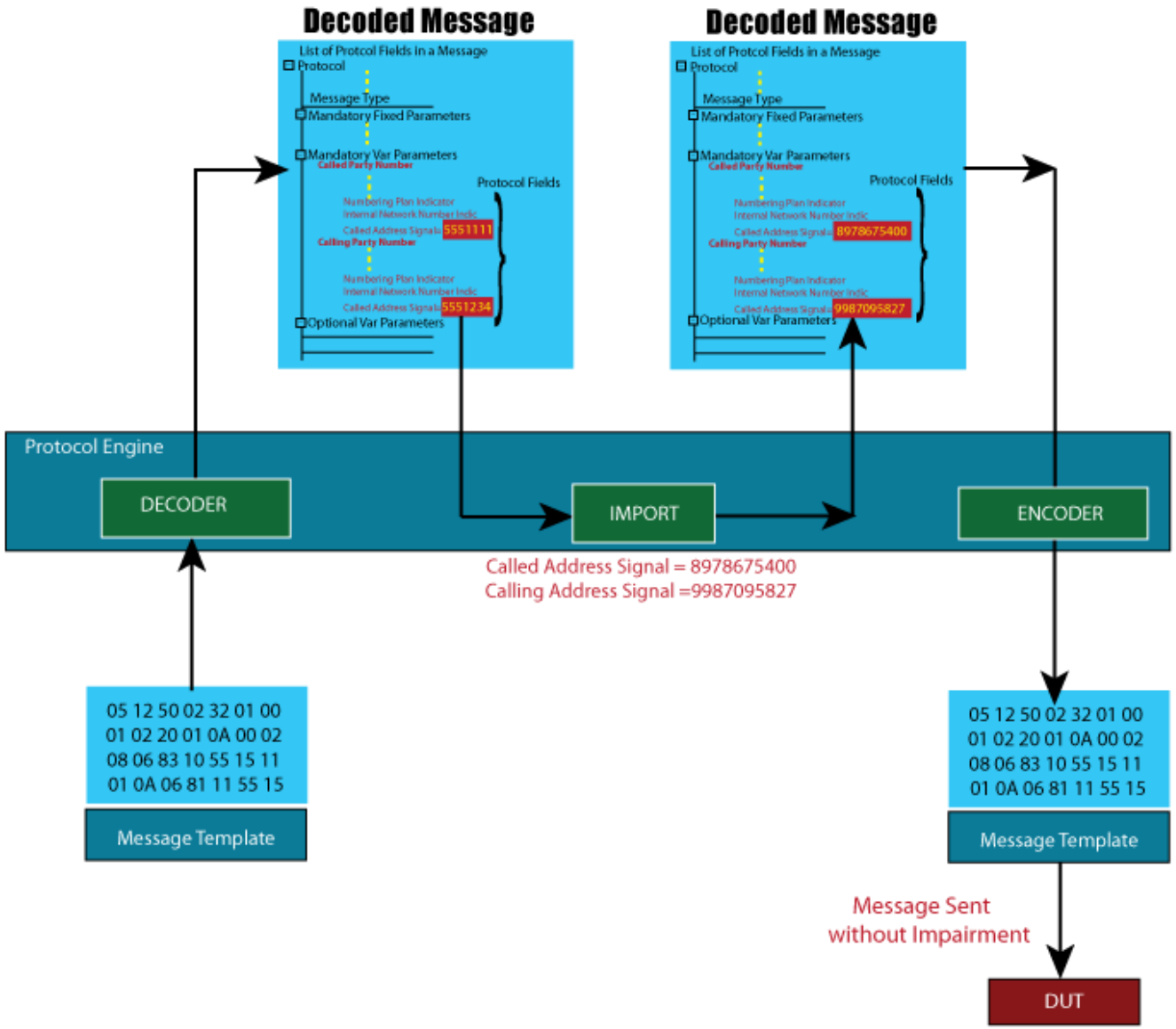
Sample Script



GL Communications

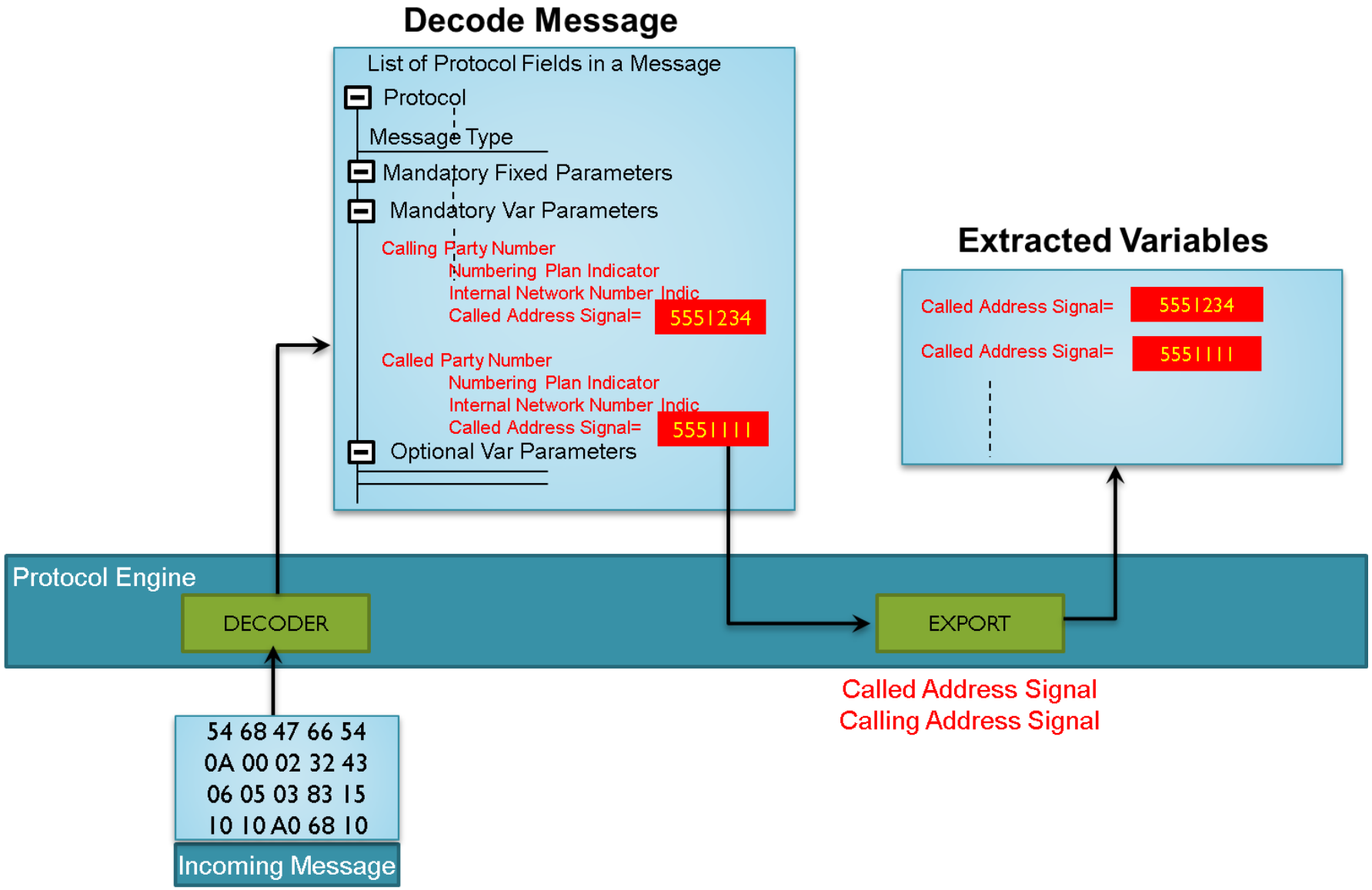
Understanding Send and Receive Messages

Basic Send Command



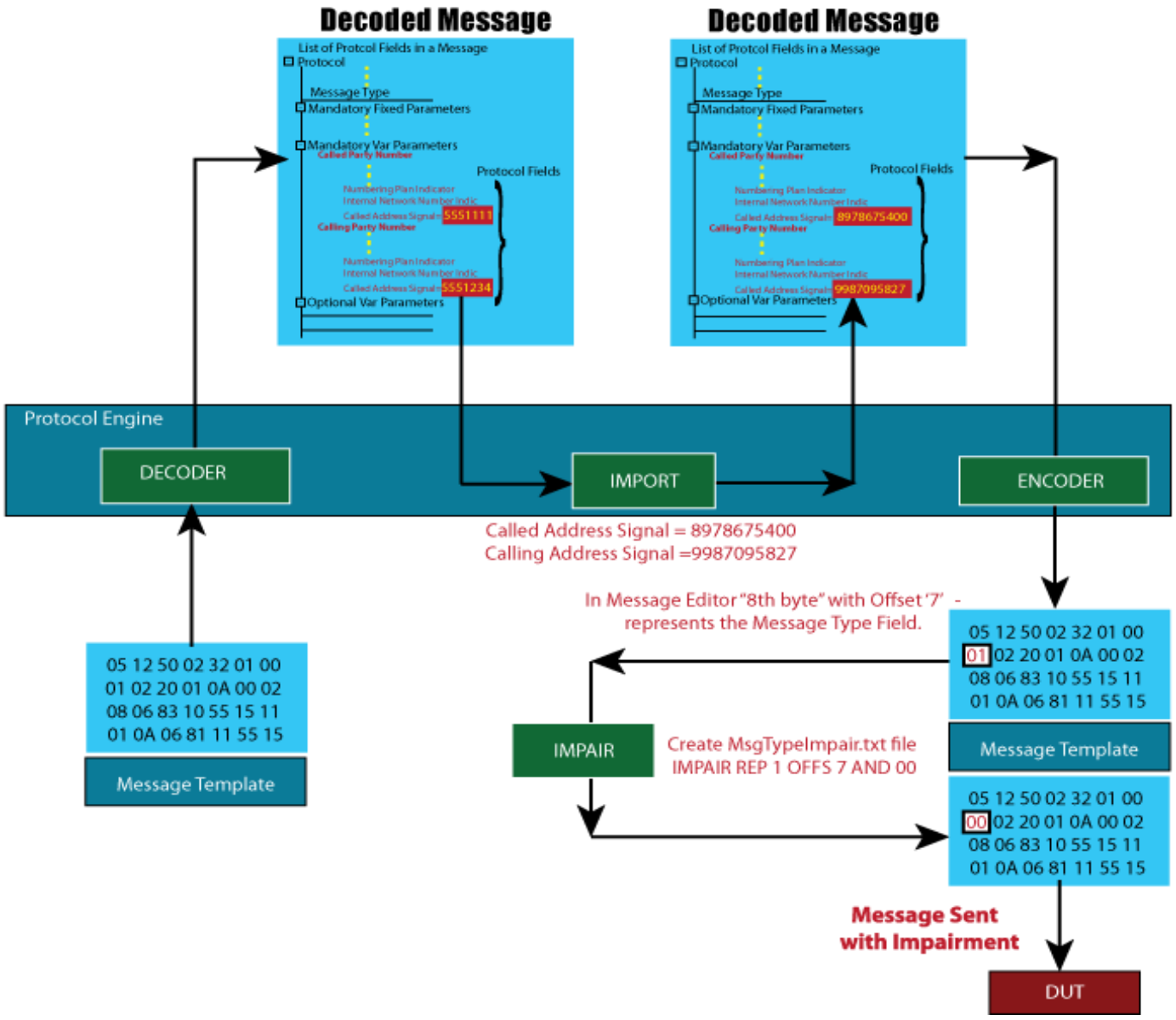
GL Communications

Basic Receive/Decode Command



GL Communications

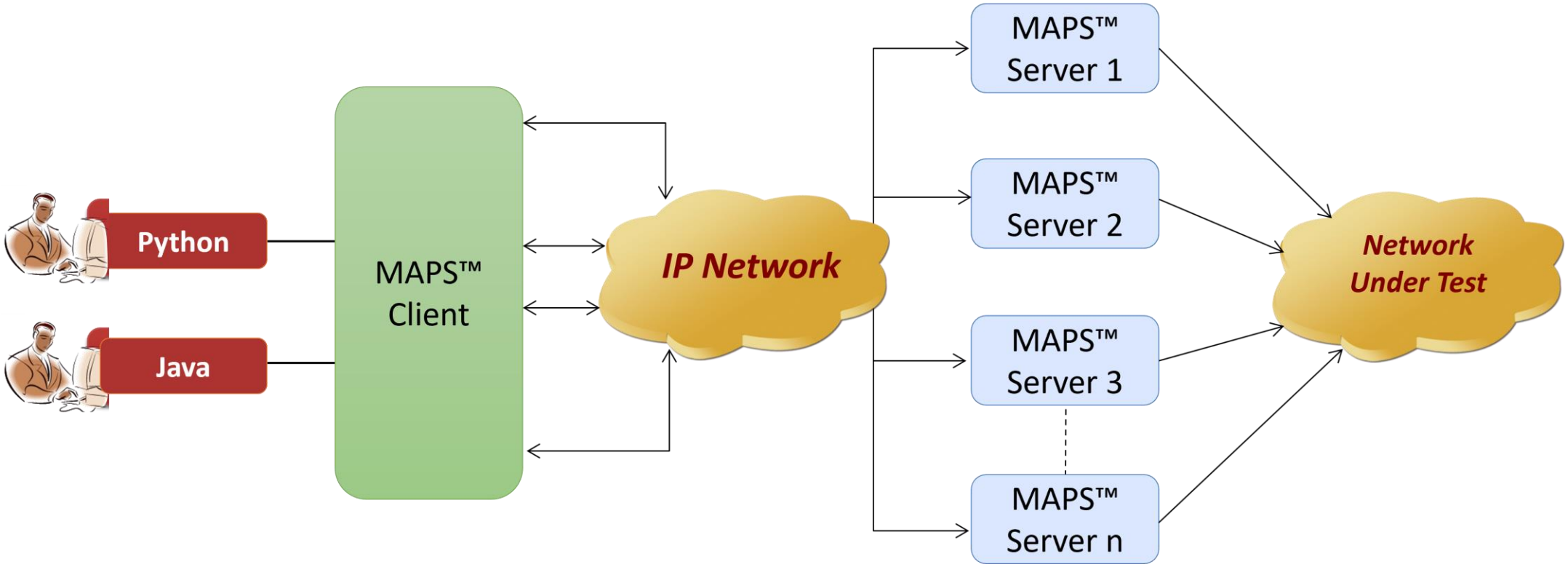
Send Command With Impairment



GL Communications

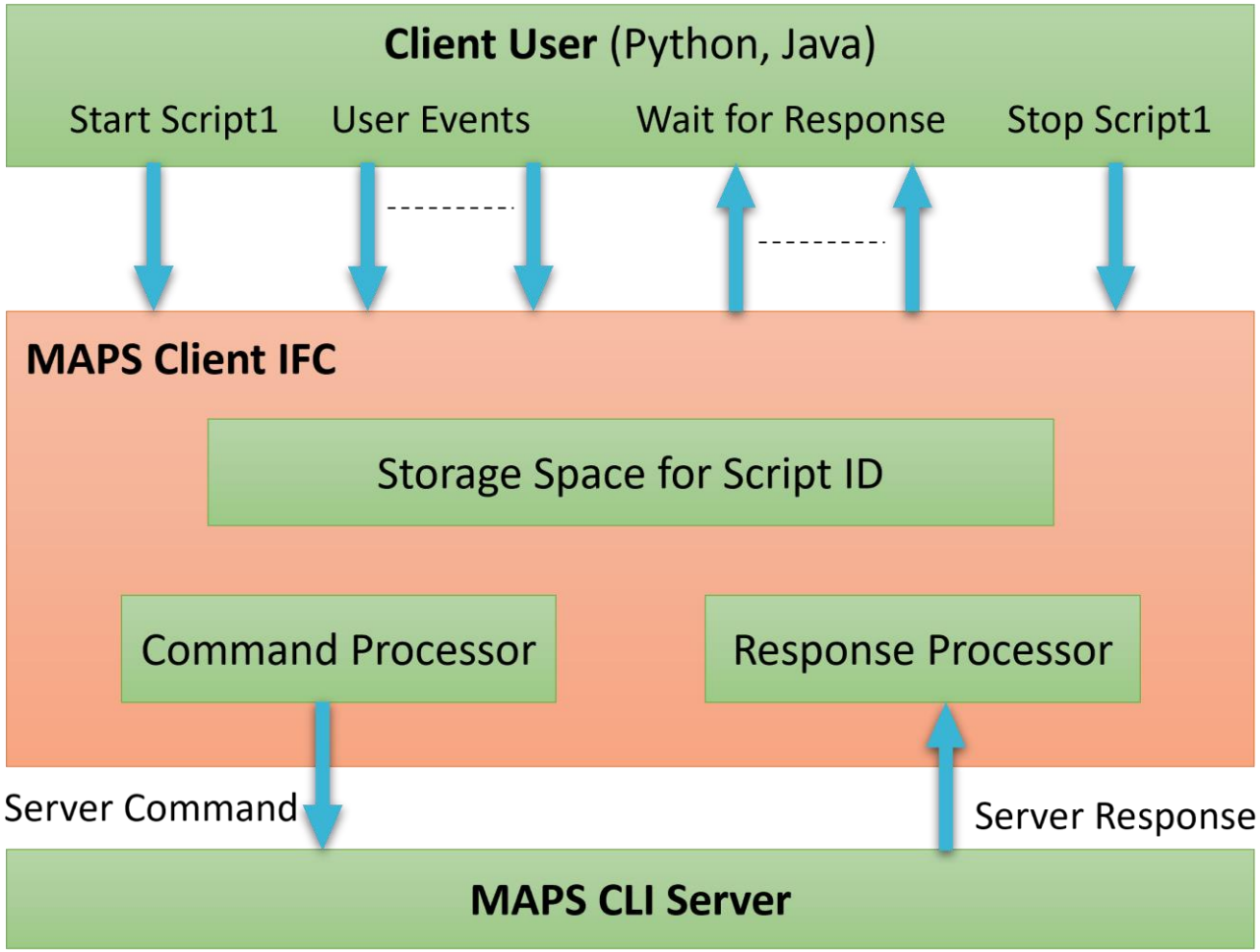
CLI/APIs for Remote Control and Test Automation

CLI for Remote Call Control and 3rd Party Integration



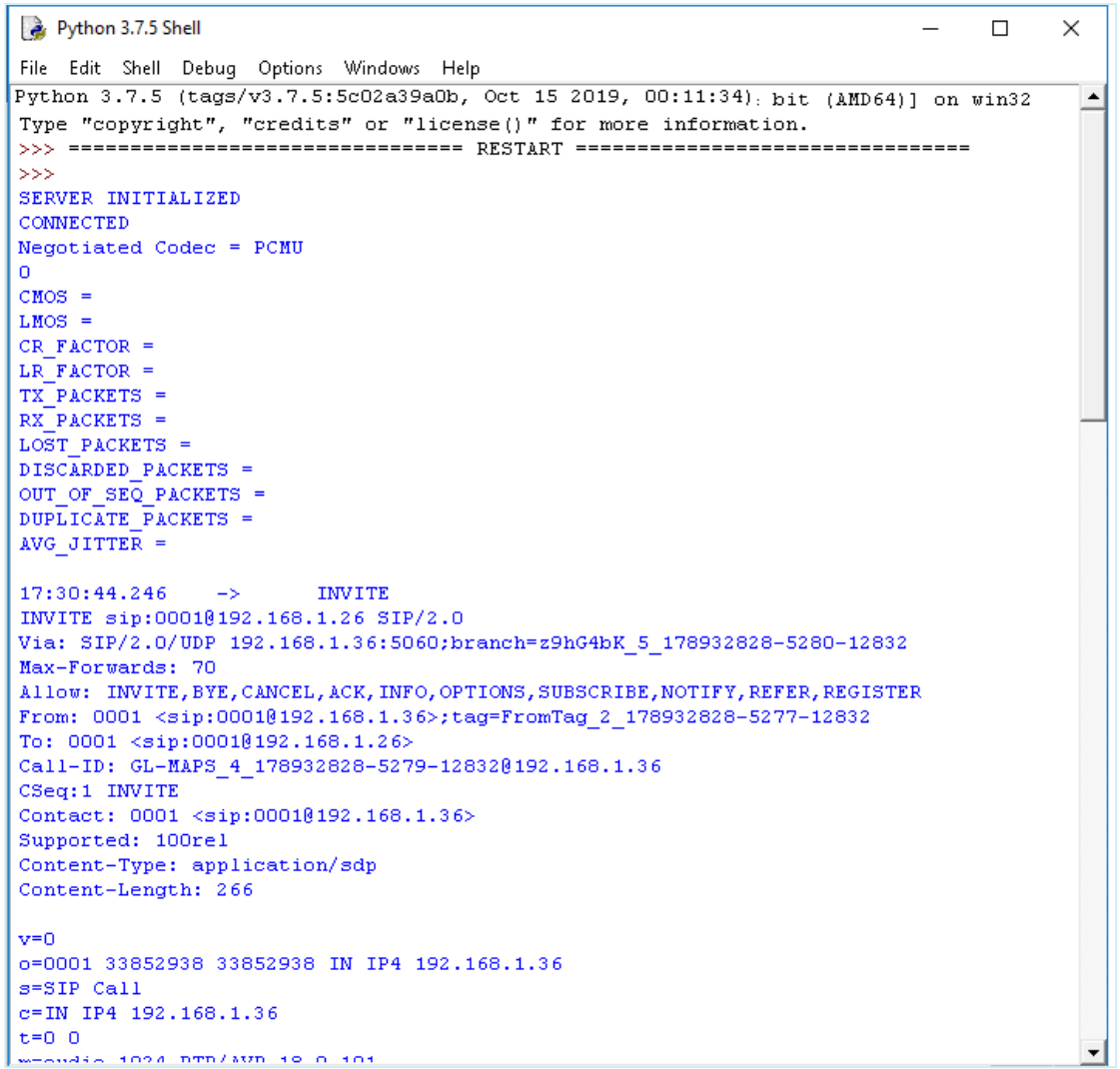
GL Communications

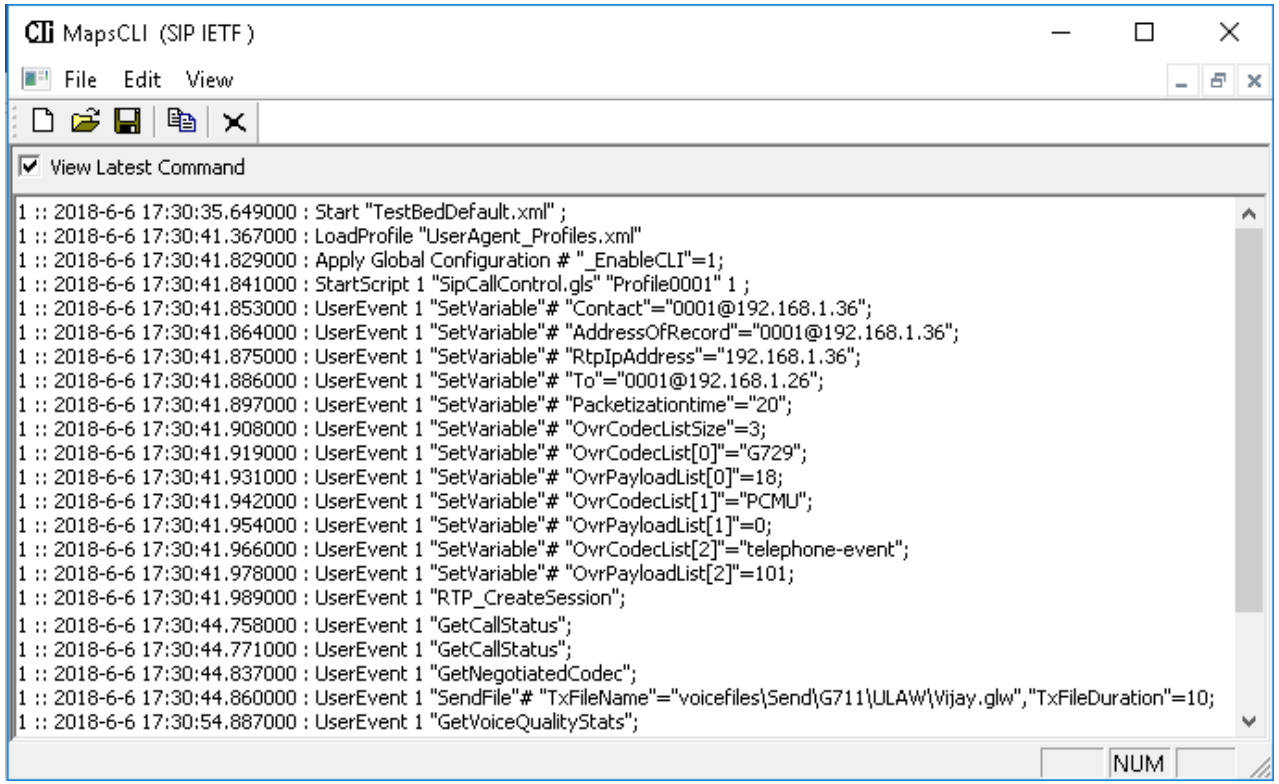
Command Line Interface Working Principle



GL Communications

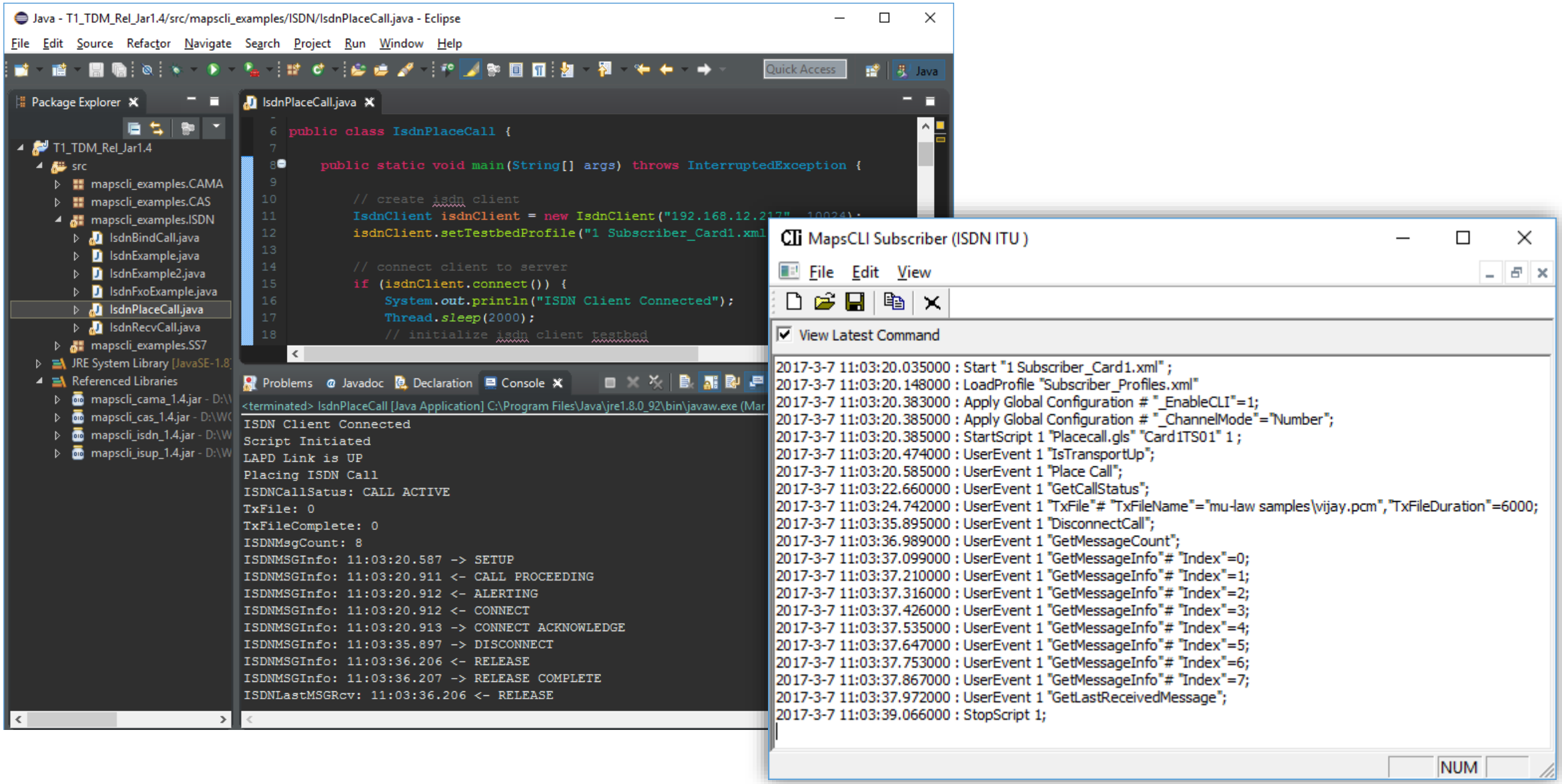
Python Client





GL Communications

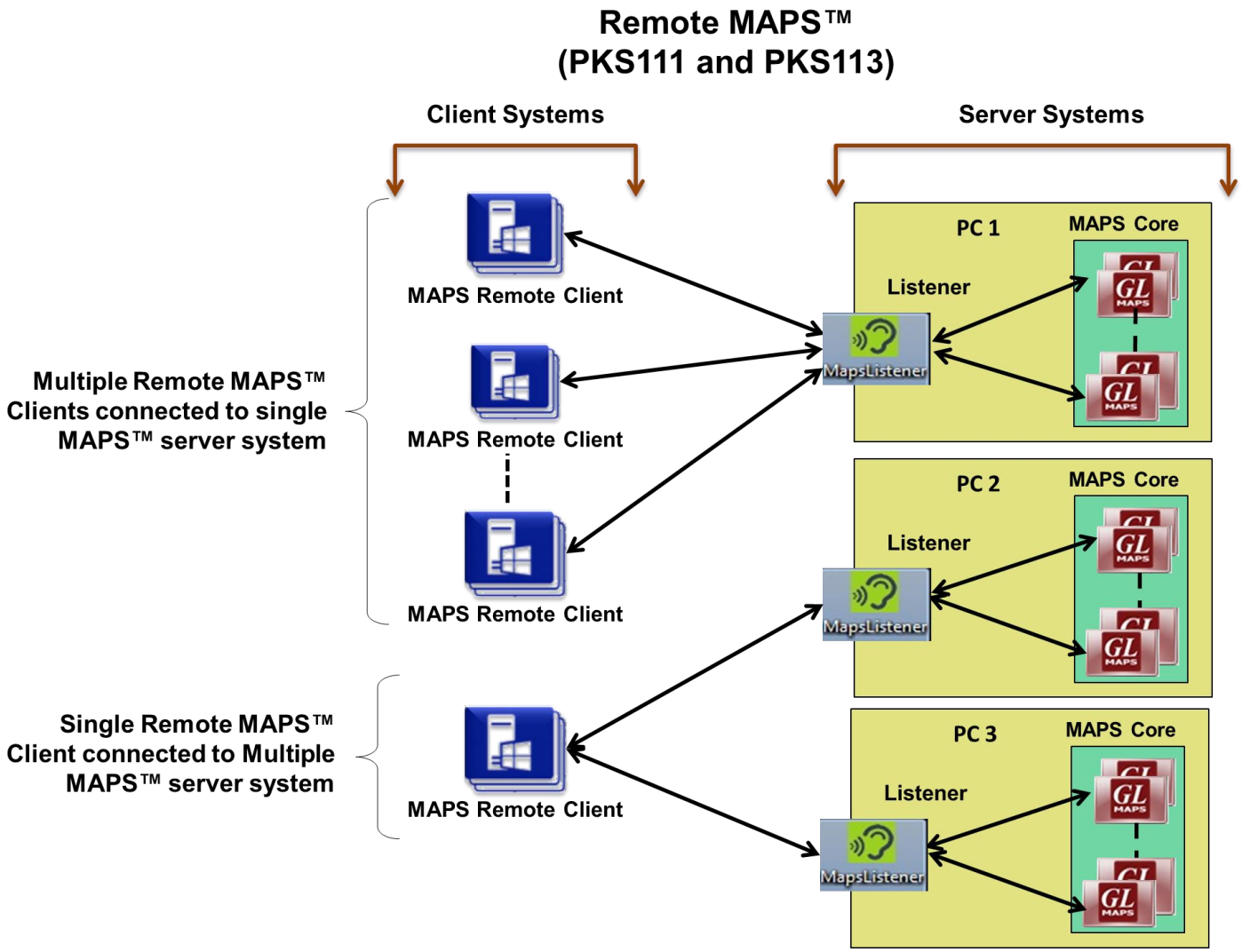
Java Client



GL Communications

Remote MAPS™ Server

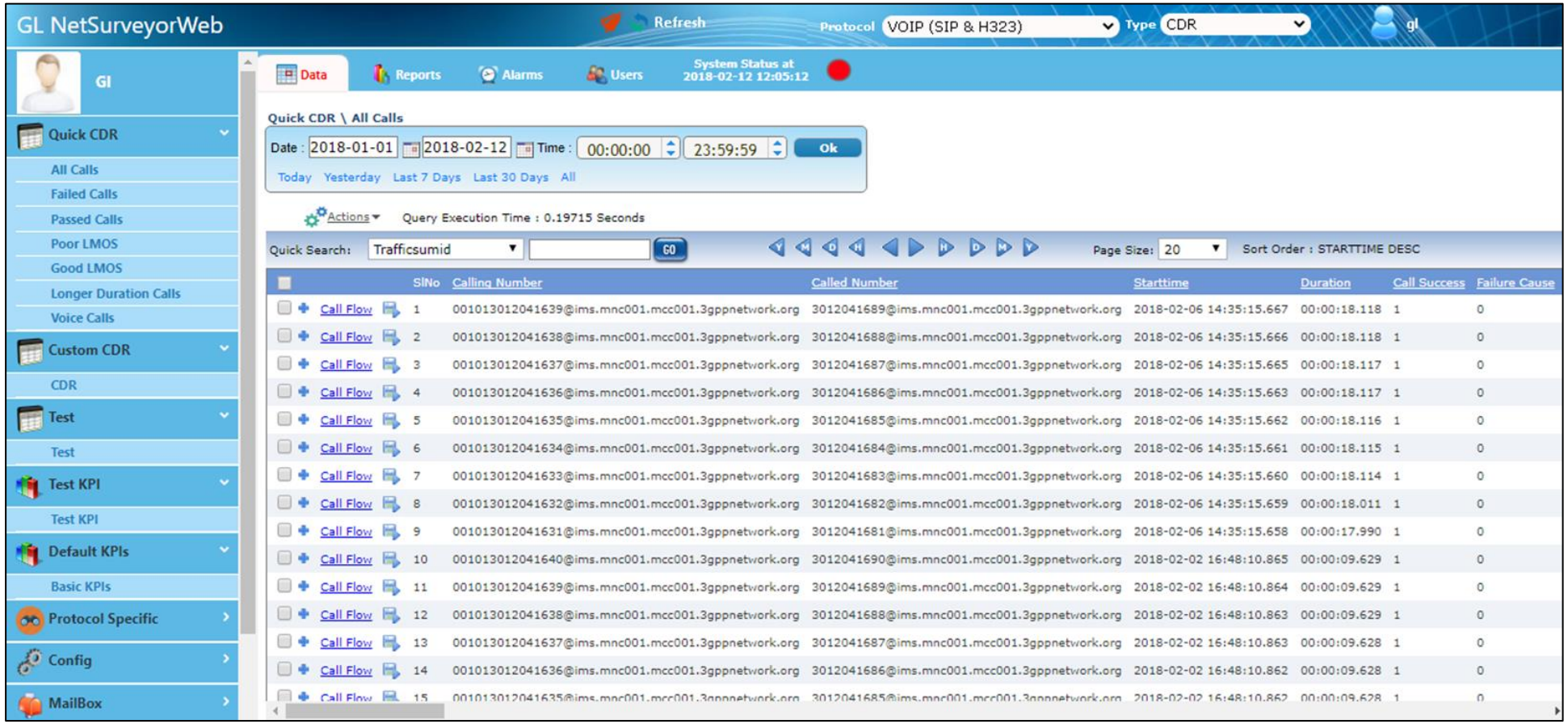
- Multi-node and multi-interface simulation from a single GUI
- Suitable for testing any core network, access network, and inter-operability functions
- Single Licensing Server controlling server and client licenses (no. of users)
- Unlimited number of remote client user can be defined at the server
- Admin privileges to control Testbed and access to configuration files for each remote client user
- Remote Client users has privileges to perform all other functions - call emulation, edit scripts/profiles, and view statistics
- Option to license multiple clients either at Remote client systems (MAPS™ Remote Client to control one or more MAPS™ Server - PKS111) or at the MAPS™ Server systems (MAPS™ Server with Multi-user capability - PKS113)
- Simultaneous traffic generation/reception at 100% on all servers



GL Communications

Send Reports to Database

- MAPS™ generated reports can be sent to Database using built in commands
- This helps to monitor and analyze test Remotely



GL Communications

Thank you