Kamailio® - Variables and Transformations

Henning Westerholt

Kamailio World

September 2021 - Online

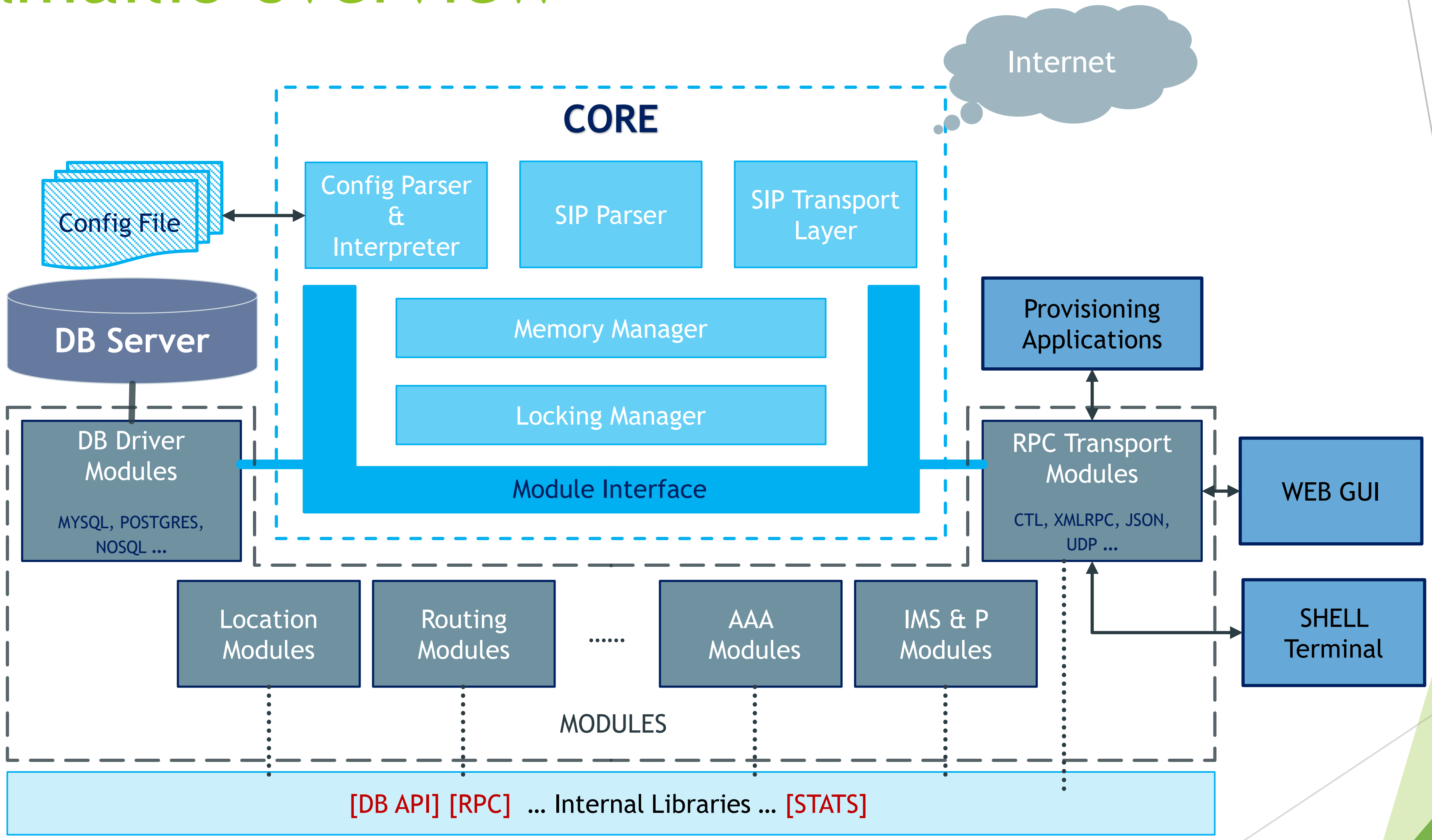
Agenda

- About
- Background
- Pseudo-variables
- Transformations
- Contact

About GILAWA

- We offer services for Real-Time Communication platforms
  - Consulting and Management
  - Administration/Developer trainings
  - Development and IT Operations
- Kamailio experience since 2007
- Independent and neutral service provider
  - No own end-user products
  - No vendor contracts
- Our customer are Internet Service Providers and Telephone Provider
- Mainly in Germany, Europe and North-America

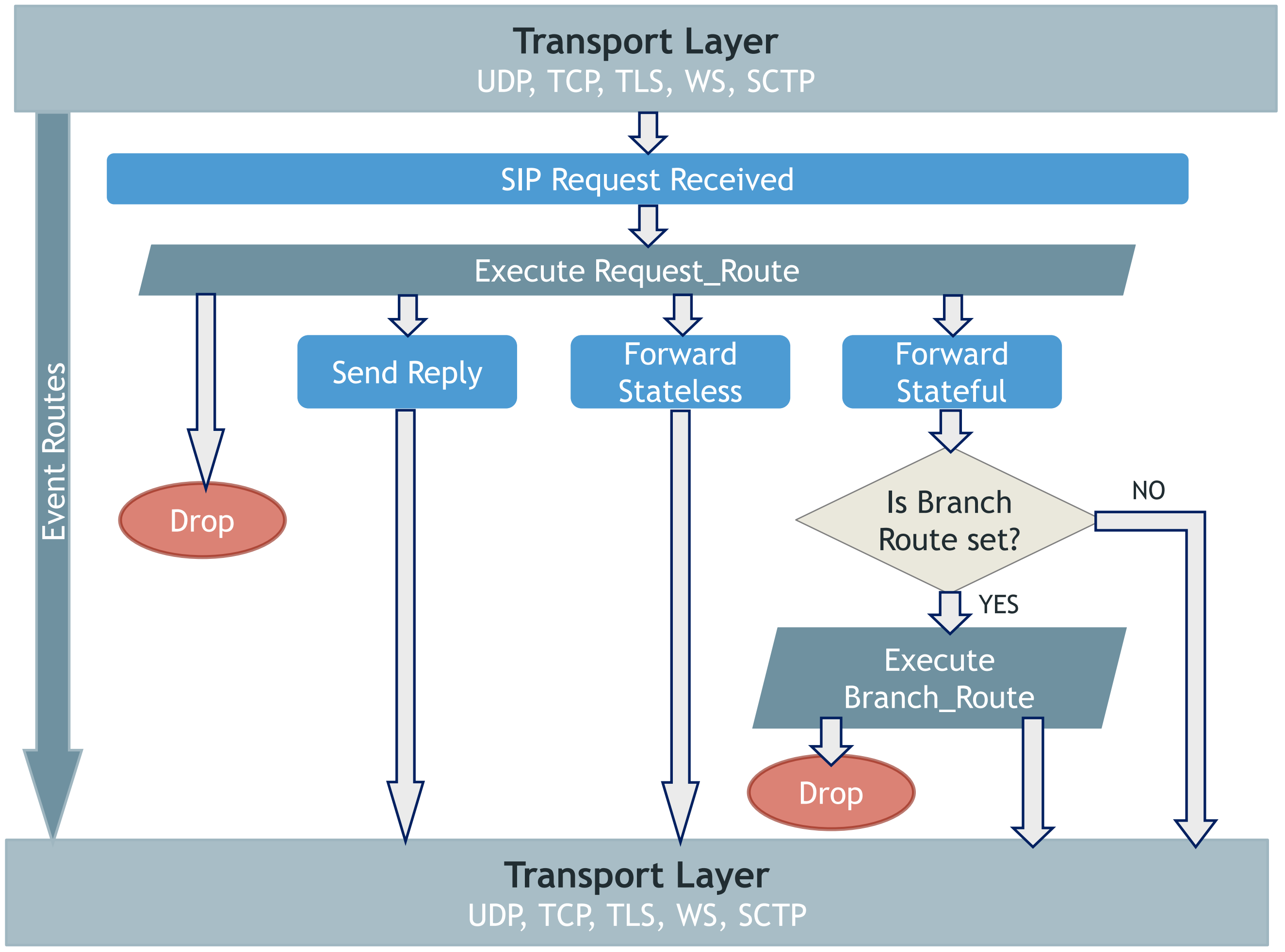
Kamailio overview



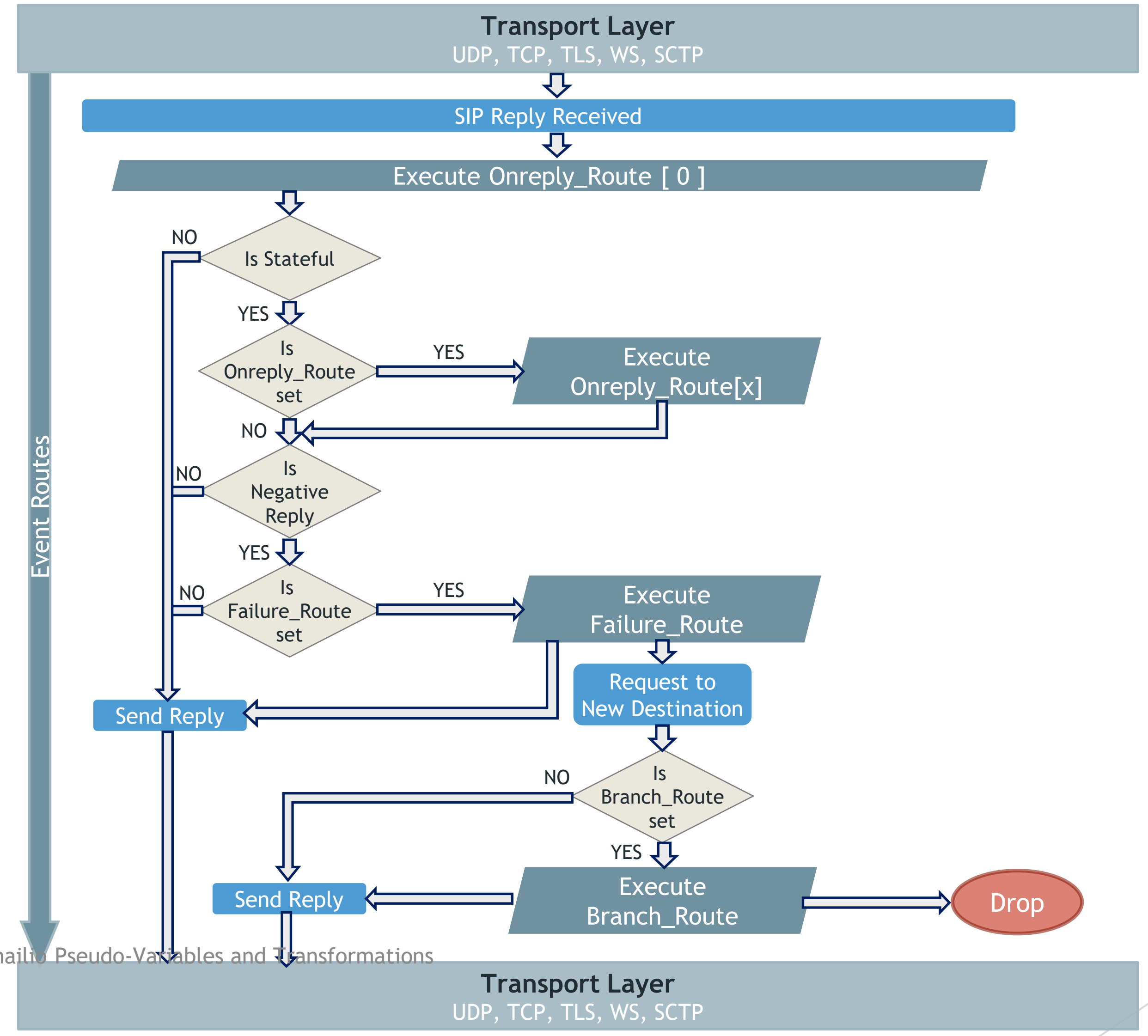
Kamailio configuration structure

- The Kamailio configuration can be structured in different main sections
  - Pre-Processing definitions and includes
  - Global options
  - Module loading
  - Module parameter
  - Main request route
  - Additional routes
- Pseudo-Variables and Transformations are mainly used in the configuration routes

SIP request processing



SIP reply processing



Small history

- Pseudo-Variables were created to generalize and standardize the existing „special“ variables in Kamailio configuration
  - Before variables would look similar to other key words of the scripting language
  - Examples: “uri” to access request URI, “src_ip” to access source IP
- First PVs were included in OpenSER 0.9.4 (2005) and extended substantially
- Kamailio 1.5.0 moved PVs (and transformations) from the core to pv module
- Transformations were added to provide a modular and extensible way to access certain information and to format them in a certain way
- They were added a bit later for OpenSER 1.2.0 (2007)
- Kamailio 1.5.0 (2009) introduced the first main generic module user - htable

What are pseudo-variables

- The term “pseudo-variable” is used for special tokens that can be given as parameters to different script functions
- They will be replaced with a value before the execution of the function
- The majority of PVs are read-only, but many of them are also read-write
- The beginning of a “pseudo-variable” is marked by the character “$”
- Pseudo-variables are implemented by various modules, most of them are provided by the „pv“ module
- They should be supported by most core and module script functions
- Usually all of them are documented in the wiki for the different versions
    - E.g., https://www.kamailio.org/wiki/cookbooks/5.5.x/
- Note about performance

Pseudo-variables usage

- From default Kamailio configuration
- if(is_method("INVITE")) {

```
 if($avp(oexten)==$null) return;
 $ru = "sip:" + $avp(oexten) + "@" + $sel(cfg_get.voicemail.srv_ip)
        + ":" + $sel(cfg_get.voicemail.srv_port);
 } else {
        if($rU==$null) return;
        $ru = "sip:" + $rU + "@" + $sel(cfg_get.voicemail.srv_ip)
               + ":" + $sel(cfg_get.voicemail.srv_port);
 }
```

Kamailio (pseudo-)variables (1/4)

- Flags (normal, branch flags) - $mf
- „Classic“ variables (uri, src_ip etc..)
- Access to SIP message content
  - $ru – request URI
  - $fU – From header user name part, $td – To header domain part
  - $hdr – access to arbitrary headers
  - There are many more, for most well-known headers are PVs available
- Access to SIP replys
  - $rr – reply reason phrase or $s – reply code

Kamailio pseudo-variables (2/4)

- Transactional
  - $avp – Attribute-Value-Pair: transactional variable
  - $xavp – extended AVP: again transactional, supports index/field access, Example: $xavp(person=>fname)=„John“; $xavp(person=>lname)=„Doe“;
  - $xavi – similar to $xavp but case insensitive for keys
  - $xavu – similar to $xavp but unique for values, no indexes
- In-memory
  - $sht – hash table (htable module)
  - $shv – variable shared between processes
  - $var – variable individual per process with default 0
  - $vn – similar to $var but with default $null

Kamailio pseudo-variables (3/4)

- Dialog stateful
  - $dlg – attributes about the processed dialog
  - $dlg_ctx – dialog context attributes about the processed dialog
  - $dlg_var – store and retrieve customer variables for the processed dialog, Example: $dlg_var(provider)=„carrier1“; if ($dlg_var(provider) == „carrier1“) {..}
- Time handling
  - $time – access to different time components
  - $TS – current unix timestamp and others
- Access to environment
  - $env – access to linux environment variables
  - $C – terminal foreground and background colors

Kamailio pseudo-variables (4/4)

- Kamailio attributes
  - $def – access defined pre-processor values
  - $version – output Kamailio version in different formats
  - $stat – return the value of statistic items
  - $mb – message buffer
- Other important module interfaces
  - $T – access to current transactions
  - $T_branch – access to current branch attributes
  - $uac_req – can be used to create SIP requests
  - $http_req – can be used to create HTTP requests

Use cases PVs

- Evaluate User-Agent string and block certain devices
  - Use $ua with regular expression match
- Increase performance for slow database operations
  - Use $sht to save DB query results
- Performance evaluations of your cfg
  - Use $TV for micro-seconds timestamp (note: benchmark module has more options)
- Multi-homed setup networks routing
  - Use $fs to specify outgoing send socket (note: now possible with send socket name)
- Access data from different sources
  - sqlops for SQL database, $redis for Redis DBs etc..

What are transformations

- A transformation is basically a function that is applied to a pseudo-variable (PV) to get a property of it
- The value of PV is not affected at all, they are read-only operations
- Provide a modular system, to prevent the addition of hundreds of special PVs
- Transformations are implemented by various modules, most of them being in pv module
- Transformations are intended to facilitate access to different attributes of PV
- A transformation is represented in between '{' and '}' and follows the PV name
- When using transformations, the PV name and transformations must be enclosed in between '(' and ')', following the $ sign

Other ways of evaluating variables

- There exists of course other methods to evaluate (pseudo-)variables
- Like textops and texpopsx, siputils modules etc..
- From SER we also inherited the „select“ Framework
- They can be accessed with $sel
- Selects provide a sub-set of PV functionality
- They are not further extended, but used in a few places

Transformation usage

- Length of From URI
  - $(fu{s.len})
- Several transformation can be applied the same time to a PV
  - Length of escaped ‘X-SBC' header body
  - $(hdr(X-SBC){s.escape.common}{s.len})

Useful Transformations (1/3)

- String operations
  - {s.len}, {s.int}, {s.trim}
- Data conversions
  - {s.encode.7bit}, {s.encode.base64url}
- Escaping
  - {s.escape.common}, {s.escape.user} (note: there is also {sql.val} available)
- URI-transformations, to access all kind of data from a SIP URI
  - {uri.user}, {uri.host}, {uri.port} etc..
  - Transformation for special use cases available, e.g. {uri.saor}
  - $var(ouri) = "sip:alice@server.com:5060;nat=yes;transport=tcp;line=xyz";
  - $var(suri) = $(var(ouri){uri.saor}); # => "sip:alice@server.com"

Useful Transformations (2/3)

- Parameters List Transformations, to access values from a concatenated string
  - {param.value,name[, delimiter]}, {param.name,index[, delimiter]}
  - "a=1;b=2;c=3"{param.value,c} = "3"
- Name-address Transformations, to access values from a "Display name" URI
- To-Body Transformations, to access values from more complex headers like To
  - {tobody.params} – the parameters can be then evaluated further
- HTTP URL Transformation
  - {url.path}, {url.querystring}
- URI Alias Transformations
  - {urialias.encode}, {urialias.decode}

Useful Transformations (3/3)

- JSON Transformation
  - {json.parse}
- Hash Transformations
  - {s.sha256}, {s.md5}
- Select operations
  - {s.select,index,separator}
- Regular expression
  - {re.subst,expression}
  - # Assign Request-URI user to PV, where every 'A' has been replaced by 'a'
  - $var(user) = $(rU{re.subst,/A/a/g});

GILAWA

Thank you

Henning Westerholt

hw@gilawa.com

https://gilawa.com/