Jasmin - Java Script MIB Implementation

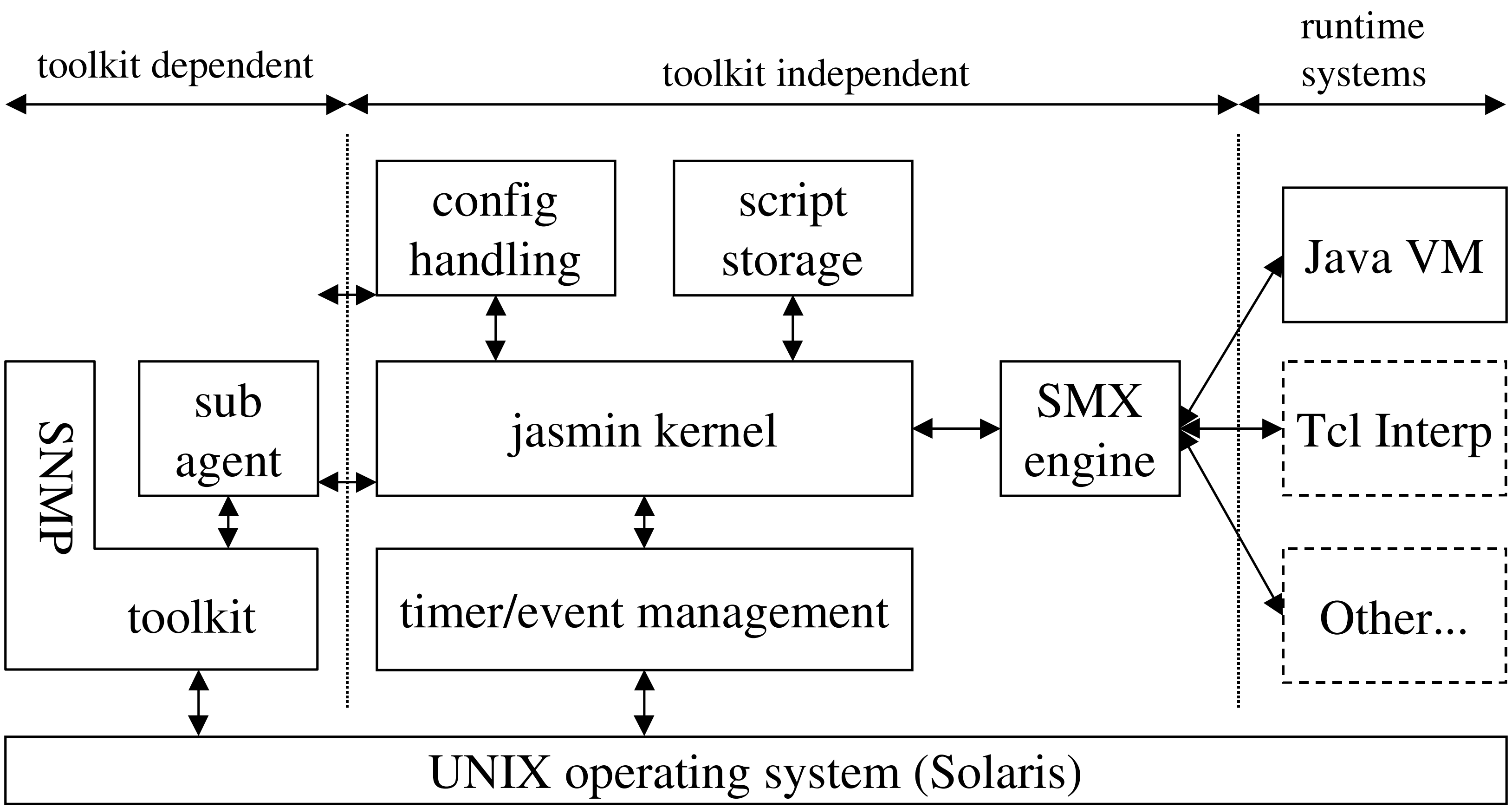
- Goals:
  - Evaluate the current Script MIB proposal by implementing and using it.
  - Use the Java virtual machine as the primary runtime system.
  - Create an architecture which allows to plug-in alternate runtime systems at low cost in order to experiment with various languages.
  - Try to adhere to the MIB specification as close as possible.
  - Share implementation experiences with the DISMAN working group.

- Participants:
  - J. Quittek, A. Kind, J. Nicklisch
    (C&C Research Labs, NEC Europe Ltd., Berlin, Germany)

  - J. Schönwälder, M. Bolz, S. Mertens
    (Technical University Braunschweig, Germany)

- Additional Information:
  - http://www.ibr.cs.tu-bs.de/projects/jasmin/

Jasmin Internal Structure



- The SMX protocol separates runtime systems (which are actually executing scripts) from the SNMP agent implementing the Script MIB.

SMX Protocol Characteristics

- Runs over a local TCP connection (accessible from all popular languages).
- Uses simple ASCII encodings and is similar to protocols like SMTP, HTTP.
- Assumes the existence of a shared file system between the SMX peers.
- The SMX peer inside the SNMP agent sends commands to the runtime system.
- The runtime system responds to commands or sends asynchronous messages.
- The SMX protocol has been designed to be simple (small number of commands and replies, simple encodings) in order to make it easy to create new runtime systems.
- Enforcement of the Script MIB integrity rules and security constraints is the responsibility of the agent. (This reduces the complexity of the runtime systems, but it requires a trusted communication channel.)
- The format of the messages is defined using the ABNF (RFC 2234).

SMX Protocol Details

- The SMX protocol currently defines 6 commands (hello, start, suspend, resume, abort, status) which are send by the SNMP agent.
- The SMX protocol currently defines 15 reply codes (grouped into classes and categories, similar to SMTP and HTTP) which are send by the runtime system.
- Most replies are send in response to a command received from the SNMP agents. Some replies may be send without a preceding command (e.g. the indication of script termination).
- Status values returned from the runtime system map directly into the Script MIB representation.
- Strings can be passed either as quoted printable strings or as binary strings in hexadecimal representation.

Jasmin SNMP Agent

- SNMP sub-agent written in C based on the Emanate toolkit.
- Does not use the data structures generated by the MIB compiler internally in order to keep the implementation in large parts independent from the SNMP toolkit.
- About 12.000 commented lines of C/lex/yacc code for the SNMP toolkit independent MIB implementation.
- About 600 lines of modified code originally generated by the MIB compiler of the SNMP toolkit.
- Uses GNU wget in order to pull scripts from remote systems via HTTP or FTP (needed to make some minor modifications since wget did not export error codes).
- Uses a trusted binary in order to enforce an operating system security profile (setting the Unix uid/gid, root directory and resource limits)

Jasmin Java Runtime

- Based on JDK 1.1.5 and supports all JDK packages except java.applet and java.awt.*.
- Allows to execute multiple „scripts“ with the same runtime security profile in a single runtime system.
- Implements runtime system security profiles with a specialized Java security manager and Java class loader class.
- Java „scripts“ (collections of Java classes) must be packaged into jar files.
- The entry point into the jar file is either defined in the jar file manifest or it is the method main() in the class Main.
- Written in 99% pure Java (needed to add a small C function in order to gain access to environment variables of the underlying Unix process).
- 2.100 lines of commented Java code for the complete Java runtime system.
(800 lines core engine, 600 lines SMX protocol, 700 lines runtime security)

Jasmin Project Status

- Prototype implementation ready for public test:
  - Send SNMP requests to osborne.ibr.cs.tu-bs.de port 161.
  - Community „junior“ or „senior“ with SNMPv1/SNMPv2c.
  - User „junior“ or „senior“ with SNMPv3 (noAuth/noPriv, empty context).

- Currently known limitations with regard to the current Script MIB ID:
  - SNMP toolkit does not provide a way to call the isAccessAllowed() ASI.
  - The smCodeTable is currently not implemented.

- Future projects:
  - Implement a Tcl runtime system based on the SMX protocol.
  - Implement some useful real-world management scripts in order to gain more experience with the script development and debugging cycle.

- Further information:
  - Visit our Web page: http://www.ibr.cs.tu-bs.de/projects/jasmin/