



Tivoli software **IBM**

**IBM Tivoli Monitoring 6.1 firewall implementation:
KDE Gateway component**

*Concepts in
Interfaces, KDE Gateway zones, and KDE Gateway functions*

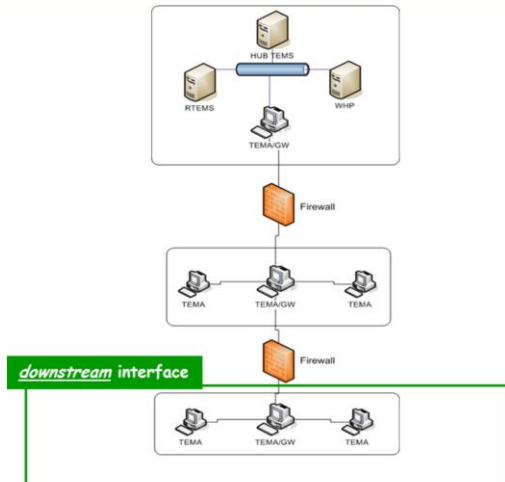
 **IBM Software Group**

IBM Tivoli Monitoring 6.1 Firewall Implementation:
KDE Gateway Component

*Concepts in
Interfaces, KDE Gateway Zones, and KDE Gateway Functions*

Concepts Interfaces

An interface describes a set of network bindings that exhibit a fixed behavior according to a specified role (upstream or downstream)



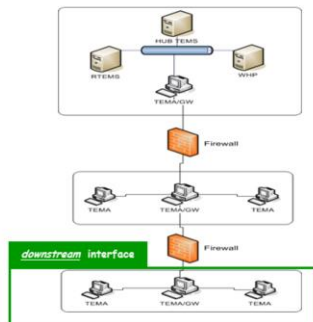
Tivoli software

IBM

Concepts

An interface describes a set of network bindings that exhibit a fixed behavior according to a specified role (upstream or downstream). You'll see here the downstream is outside the firewall.

Concepts Interfaces (continued)



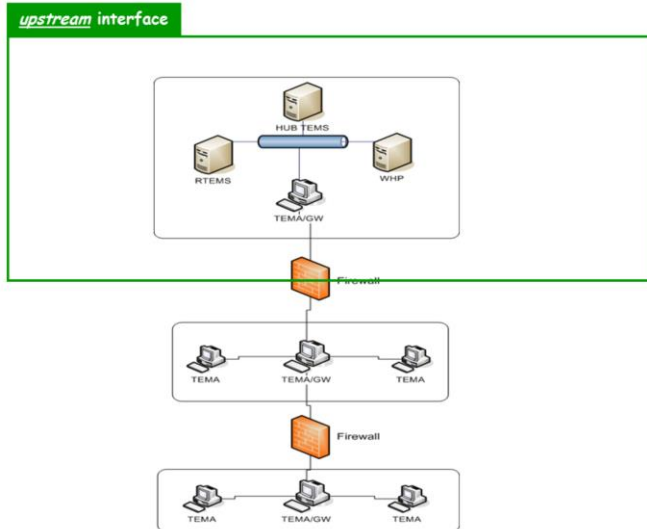
- "Outer-most" network interface is called **downstream**.
- The application client (TEMA, remote TEMS, etc) resides on the "outer-most" interface.
- Listening downstream server proxy interface

Tivoli software

IBM

Downstream is the outer-most network interface. The application layer is in the downstream. The TEMA agent (You could have a remote TEMS that is supported and works through there) resides in the outer-most. They are listening downstream server proxy interface. [Server proxy components reside downstream](#) and listen for inbound connections.

Concepts Interfaces (continued)

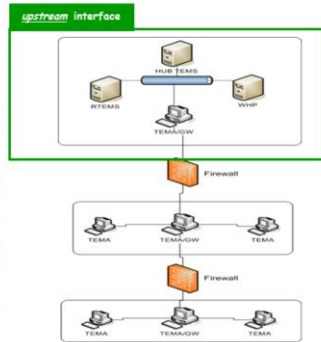


Tivoli software

IBM

Upstream is inside the firewall. It's where the HUB TEMS and the Warehouse Proxy reside.

Concepts Interfaces (continued)




- "Inner-most" network interface is called **upstream**.
- The application server (TEMS) resides on the "inner-most" interface.
- Upstream client proxy interface making the connections.

Tivoli software

IBM

Client proxy components reside upstream and make connections to services on behalf of downstream endpoints

The application server (TEMS) resides on the inner-most. Not your other application but the monitoring application server. The upstream interface is making connections.



Concepts

KDE Gateway Zones

6

A zone is a container of interfaces sharing communication resources.

- A KDE Gateway is composed of one or more *KDE Gateway zones*
- *KDE Gateway zones* are described starting with the *downstream* (or outer-most) *zone* and moving to the *upstream* (inner-most) *zone*.
- A *KDE Gateway zone* contains *one or more upstream Gateway interfaces* and their associated *downstream Gateway interfaces*.

Tivoli software

IBM

A zone is a container of interfaces sharing communication resources. A KDE Gateway is composed of one or more zones. And the zones are described starting with the downstream (or outer-most) zone and moving to the upstream (inner-most) zone. A KDE Gateway zone contains one or more upstream Gateway interfaces and their associated downstream Gateway interfaces.

7

Concepts

KDE Gateway functions

- There are two basic KDE Gateway functions:
 - > **Proxy functions:** capture socket requests from the application.
 - > **Relay functions:** establish a point-to-point link (TCP session) and relay socket requests across that link.
- Gateway functions are configured through an XML document, pointed to in the process by the KDE_GATEWAY environment variable.

The diagram shows a circle labeled 'KDE Gateway interface' containing three stacked rectangular boxes: a top green box labeled 'Relay', a middle yellow box labeled 'TEMA/TEMS/WPA', and a bottom green box labeled 'Proxy'. A callout bubble points to the 'Relay' box with the text: 'Implement Gateway via: • .xml • KDE_GATEWAY'.

A Gateway interface can assume one of three roles: listen, connect or proxy. (The proxy and relay functions discussed above are actually Gateway interfaces with specific upstream and downstream roles.)

Relay allows comms between zones

Bidirectional communication

- The destination port needs to be full duplex since the communication can flow both directions.
- A request will be sent through the port and an acknowledgement will come back so this port needs to be full duplex

There are two basic functionalities of a gateway: it's a proxy function to capture the socket request from the application, and then a relay function to establish point-to-point communication and relay the request across that link. The gateway functions are configured through an XML document, and they are pointed to by a variable, KDE_GATEWAY, and that is in the environment file of the agent which you are setting up. If you are using the UNIX agent in the ux.ini you would have a variable called KDE_GATEWAY, and it will point to the XML file. And it does need to be fully qualified. You would put in there XML file firewall, outside.xml. In that environment file, you're going to have that variable point straight to it./op/ibm/itm/where you have it, you want to point to it. You want to make sure it's the one you're expecting it to use and that that variable doesn't get removed or deleted. It's read on the start of the agent and it will pull in the settings from that XML file. One thing to point out here is this communication is bi-directional. The destination port needs to be full duplex since the communication can flow both directions. Requests are sent through the port and acknowledgements do come back. Yes the firewall port

needs to be open both ways.

Copyright and trademark information

© Copyright IBM Corporation 2000 - 2007. All rights reserved.

U.S. Government Users Restricted Rights - Use, duplication or disclosure restricted by
GSA ADP Schedule Contract with IBM Corp.

IBM web site pages may contain other proprietary notices and copyright information which
should be observed.

IBM trademarks

<http://www.ibm.com/legal/copytrade.shtml#ibm>

Fair use guidelines for use and reference of IBM trademarks

<http://www.ibm.com/legal/copytrade.shtml#fairuse>

General rules for proper reference to IBM product names

<http://www.ibm.com/legal/copytrade.shtml#general>

Special attributions

<http://www.ibm.com/legal/copytrade.shtml#section-special>

