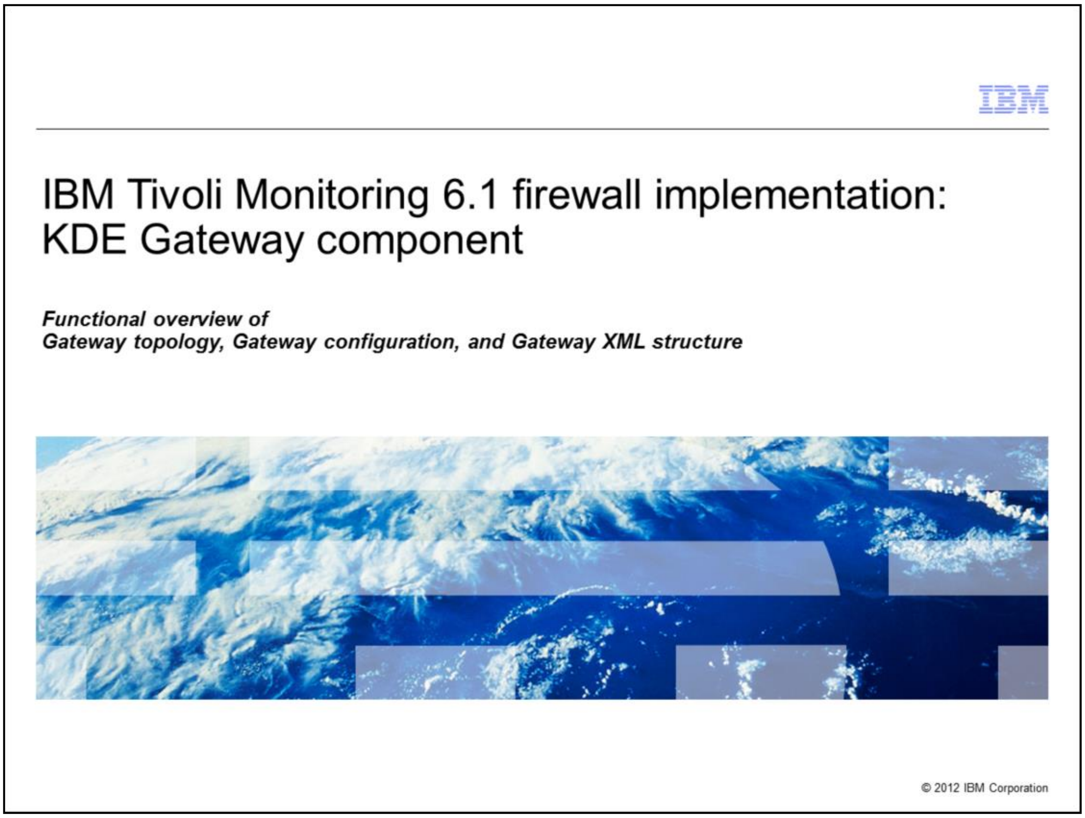
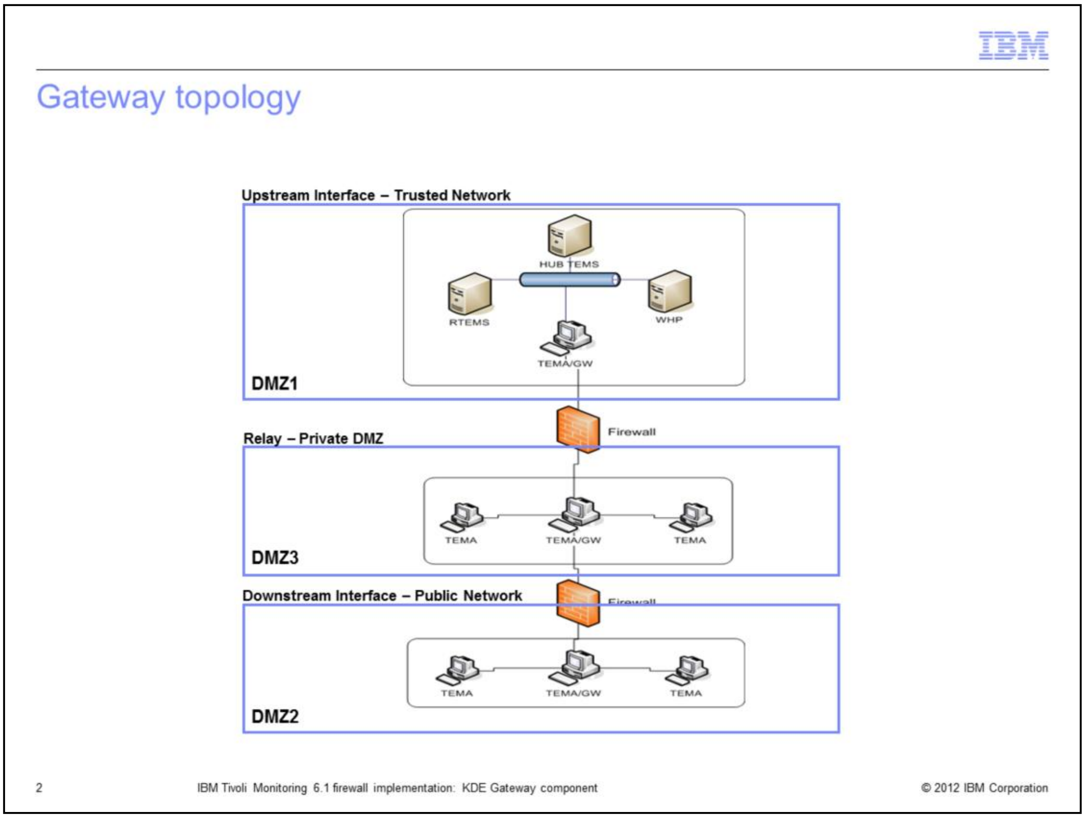


IBM Tivoli Monitoring 6.1 firewall implementation: KDE Gateway component

Functional overview of
Gateway topology, Gateway configuration, and Gateway XML structure

This presentation describes the IBM Tivoli Monitoring 6.1 Firewall Implementation: KDE Gateway Component.

Functional Overview of Gateway Topology, Gateway Configuration, and Gateway XML Structure

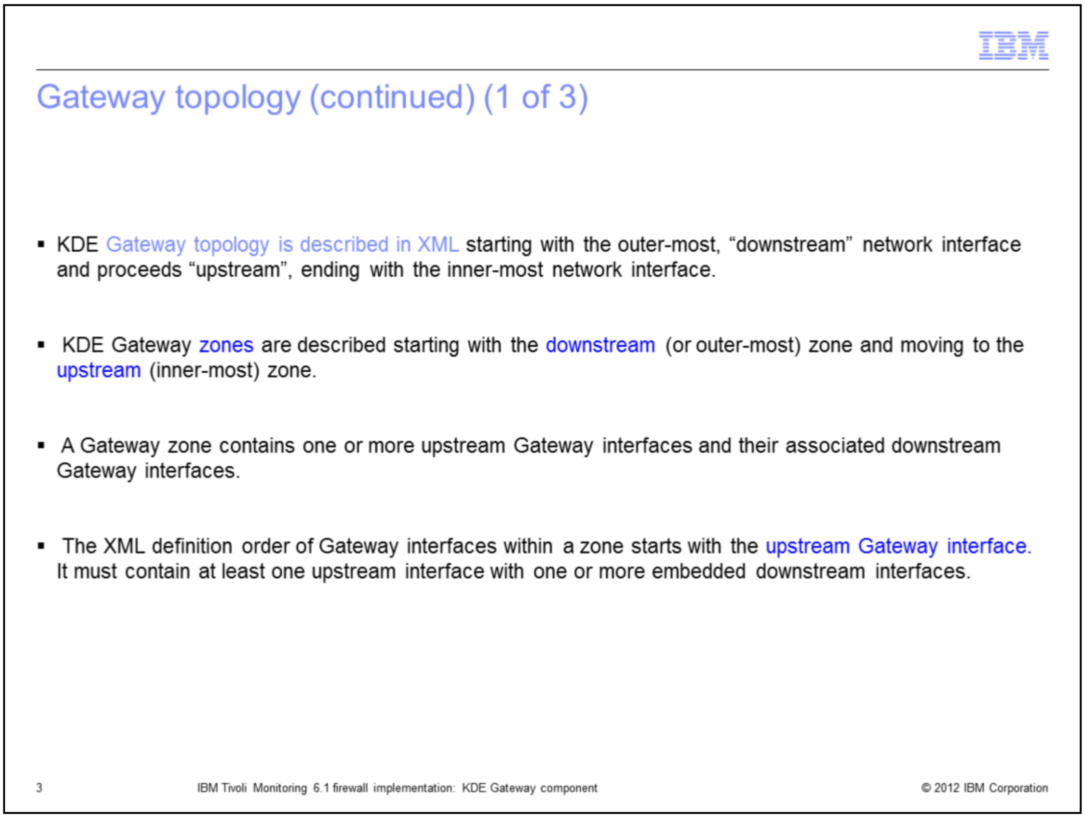


Gateway topology

2 IBM Tivoli Monitoring 6.1 firewall implementation: KDE Gateway component © 2012 IBM Corporation

KDE Transport Layer:

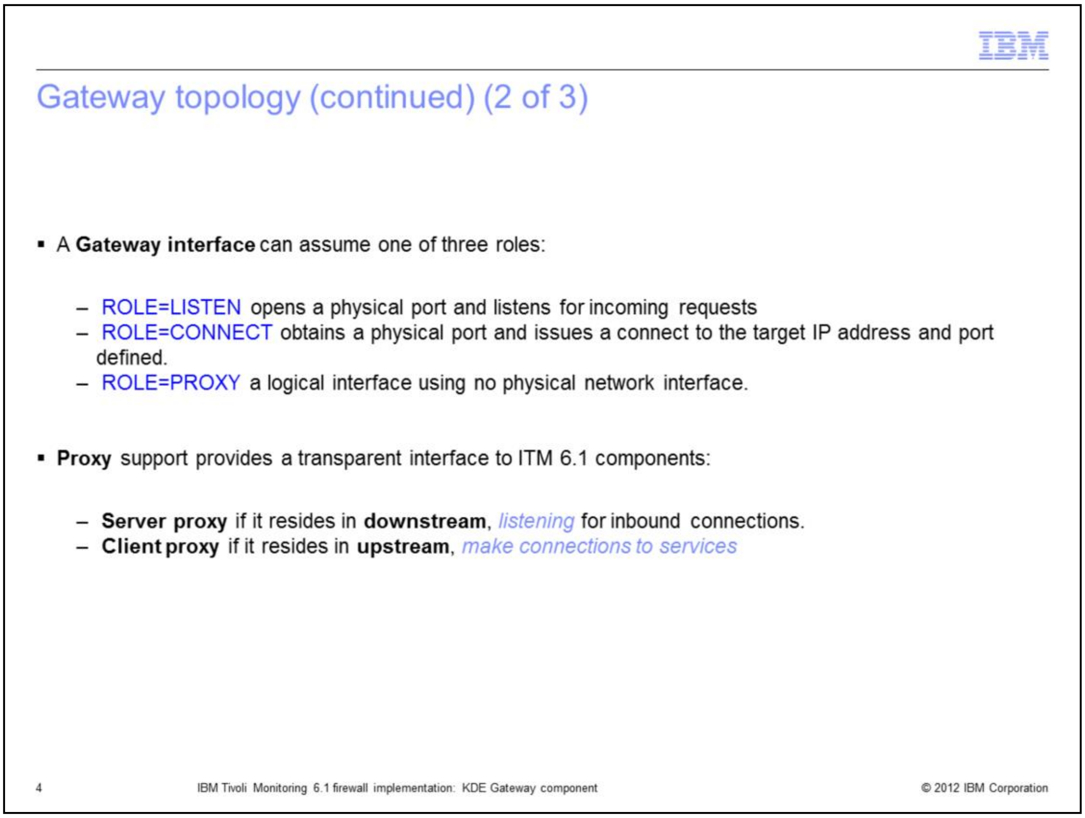
The Gateway is implemented in the KDE transport layer; it introduces a level of abstraction creating a socket-independent layer of “Monitor” calls. This is the simple topology that has been seen before, and with the definition of the interfaces. The public network is the downstream interface. And then a private DMZ zone will act as a relay. The upstream interface is the trusted network.



Gateway topology (continued) (1 of 3)

- KDE Gateway topology is described in XML starting with the outer-most, “downstream” network interface and proceeds “upstream”, ending with the inner-most network interface.
- KDE Gateway zones are described starting with the downstream (or outer-most) zone and moving to the upstream (inner-most) zone.
- A Gateway zone contains one or more upstream Gateway interfaces and their associated downstream Gateway interfaces.
- The XML definition order of Gateway interfaces within a zone starts with the upstream Gateway interface. It must contain at least one upstream interface with one or more embedded downstream interfaces.

The topology is described in XML files. To describe this topology, you need to start from the downstream network interface and proceed upwards to the inner-most network interface that is the upstream interface. The zones are described as starting with the downstream zone and moving to the upstream zone. Each gateway zone can contain one or more upstream interfaces and their associated downstream Gateway interfaces. The XML definition order of a Gateway interface within a single zone will start with the upstream interface for that zone and will contain the related downstream gateway interface for the zone that is being defined.

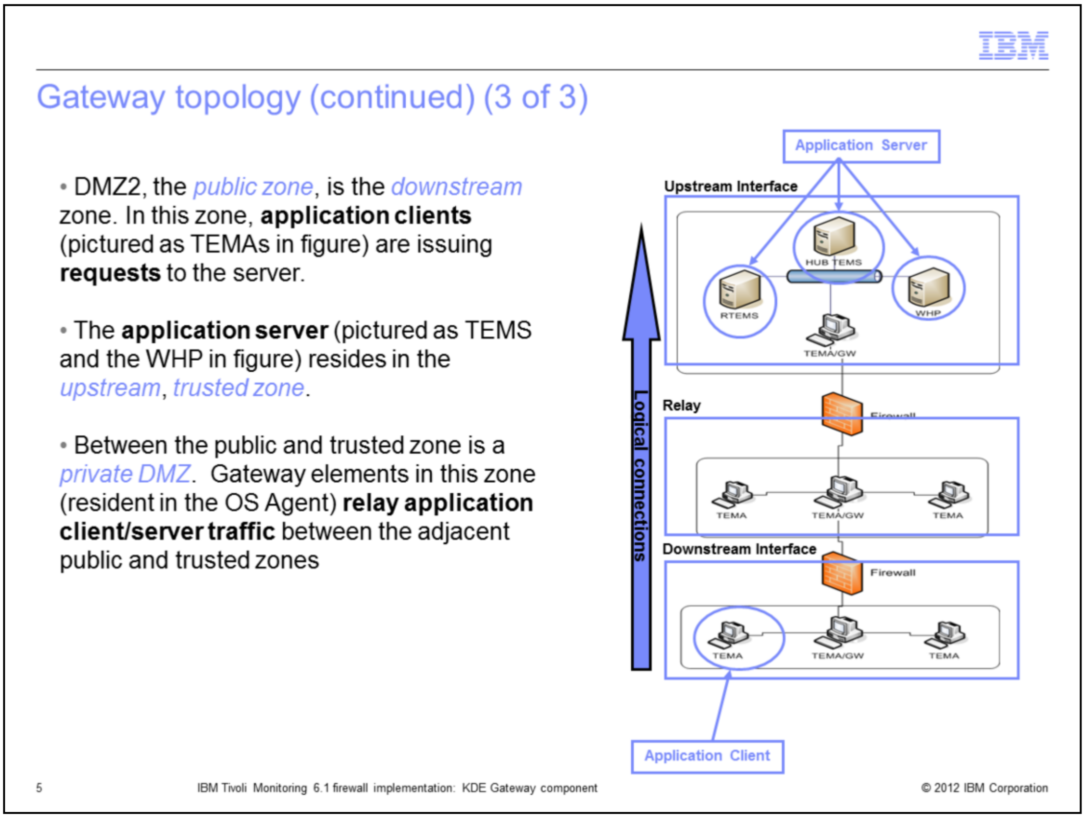


Gateway topology (continued) (2 of 3)

- A **Gateway interface** can assume one of three roles:
  - ROLE=LISTEN opens a physical port and listens for incoming requests
  - ROLE=CONNECT obtains a physical port and issues a connect to the target IP address and port defined.
  - ROLE=PROXY a logical interface using no physical network interface.
- **Proxy** support provides a transparent interface to ITM 6.1 components:
  - **Server proxy** if it resides in **downstream**, *listening* for inbound connections.
  - **Client proxy** if it resides in **upstream**, *make connections to services*

4 IBM Tivoli Monitoring 6.1 firewall implementation: KDE Gateway component © 2012 IBM Corporation

A Gateway interface can assume one of these three roles. The listen, the connect and the proxy. When you define an interface with the role equal to listen, you are basically opening a physical port, and you will listen to incoming requests on that port. If you define an interface with a role equal to connect you will obtain a physical port and will issue a connect request to the target IP address and the port defined in the XML definition. When you define the role equal to the proxy, this logical interface will not use a physical network interface. The interface is defined as the proxy with role equal to proxy. If it is a resident in the downstream interface. It is listening for inbound connections, in your case from the agents. A client proxy is defined as an interface that is a resident in the upstream interface and defined with role equal to proxy. This interface will make the connections to the services. The services here are the request to TEMS or the request to access Warehouse Proxy data.



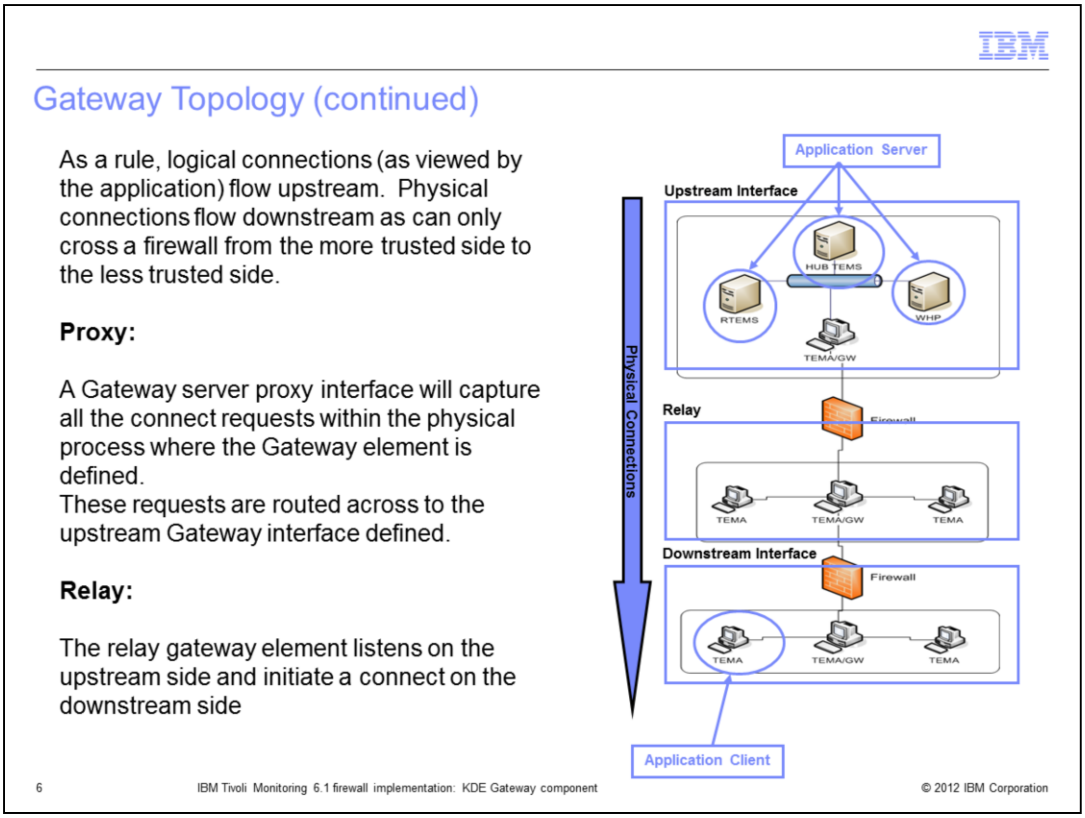
Gateway topology (continued) (3 of 3)

- DMZ2, the *public zone*, is the *downstream* zone. In this zone, **application clients** (pictured as TEMAs in figure) are issuing **requests** to the server.
- The **application server** (pictured as TEMS and the WHP in figure) resides in the *upstream*, *trusted zone*.
- Between the public and trusted zone is a *private DMZ*. Gateway elements in this zone (resident in the OS Agent) **relay application client/server traffic** between the adjacent public and trusted zones

5 IBM Tivoli Monitoring 6.1 firewall implementation: KDE Gateway component

The public zone is defined as the downstream zone. In this zone, the applications clients like TEMAs for example, are issuing requests to the server. The application server that is pictured as TEMS and Warehouse Proxy agent in the figure resides in the upstream zone, in the trusted network.

Between the public and the trusted zone there is a private DMZ. The gateway elements in this zone relay application client/server traffic between the adjacent public and trusted zones.



Gateway Topology (continued)

As a rule, logical connections (as viewed by the application) flow upstream. Physical connections flow downstream as can only cross a firewall from the more trusted side to the less trusted side.

Proxy:

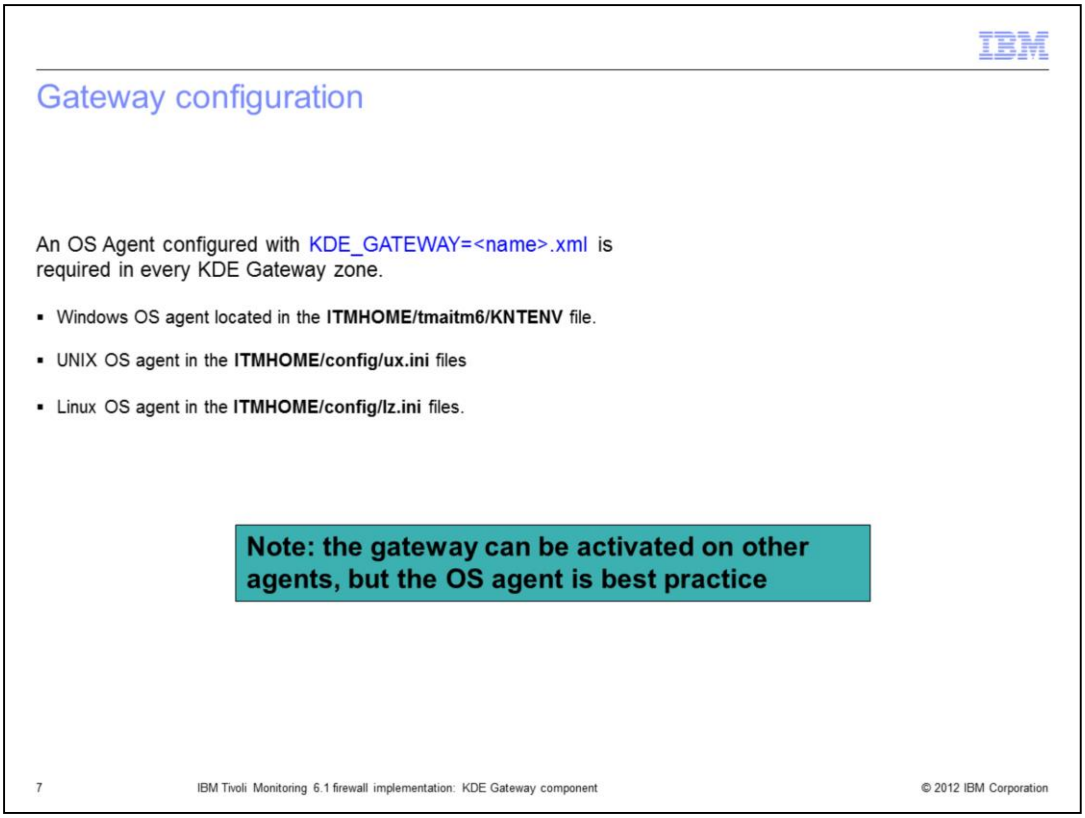
A Gateway server proxy interface will capture all the connect requests within the physical process where the Gateway element is defined.
These requests are routed across to the upstream Gateway interface defined.

Relay:

The relay gateway element listens on the upstream side and initiate a connect on the downstream side

The physical connections are defined from downstream, from the more trusted side to the less trusted side, because this is a rule of a firewall. A gateway server proxy interface will capture all the connect requests within the physical process where the Gateway element is defined. These requests are routed across to the upstream Gateway interface.

The relay gateway element listens on the upstream side and initiates a connect on the downstream side. The XML definition is described in a later slide.



Gateway configuration

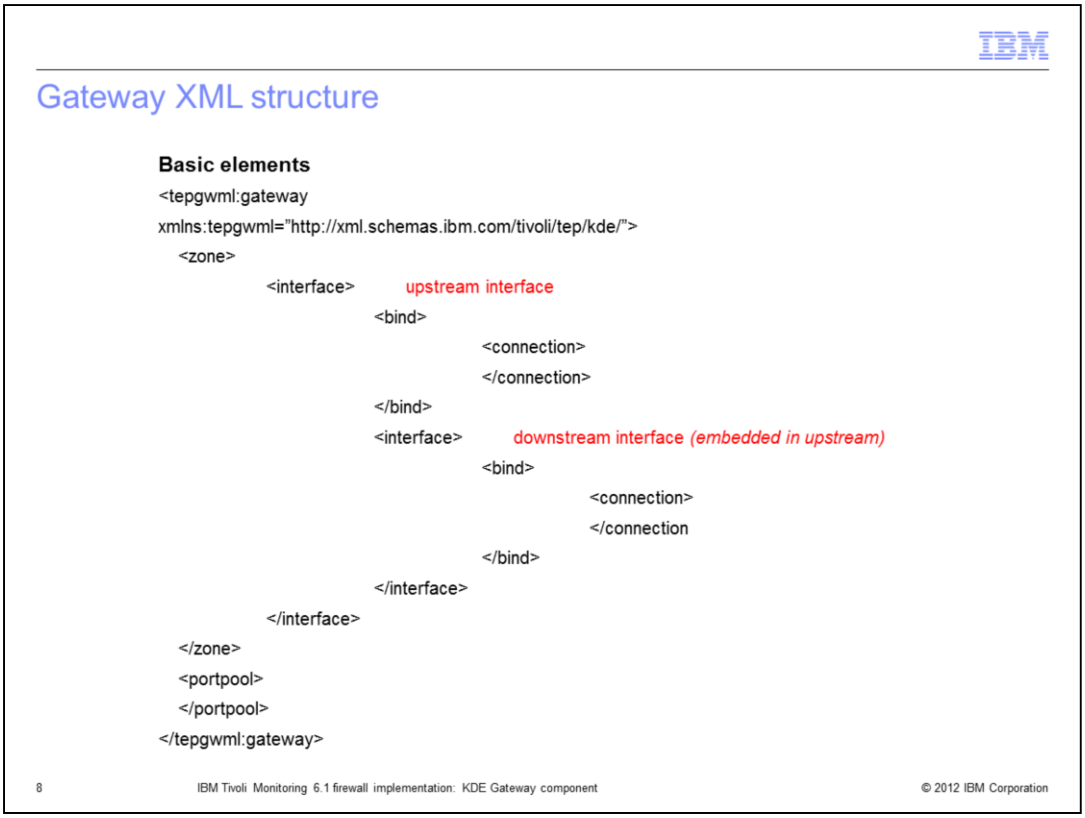
An OS Agent configured with KDE_GATEWAY=<name>.xml is required in every KDE Gateway zone.

- Windows OS agent located in the **ITMHOME/tmaitm6/KNTENV** file.
- UNIX OS agent in the **ITMHOME/config/ux.ini** files
- Linux OS agent in the **ITMHOME/config/lz.ini** files.

Note: the gateway can be activated on other agents, but the OS agent is best practice

7 IBM Tivoli Monitoring 6.1 firewall implementation: KDE Gateway component

Here is how and where to configure an OS Agent, for example, to implement the KDE Gateway feature. You will have to define the variable KDE_GATEWAY equal to the name of the XML that will define the gateway topology. This variable must be contained in the environment files, depending on the platform and the OS agent that is used. The gateway can be activated on other agents, for example universal agents, but the OS agent is the best practice and is the suggested agent to be used.



Gateway XML structure

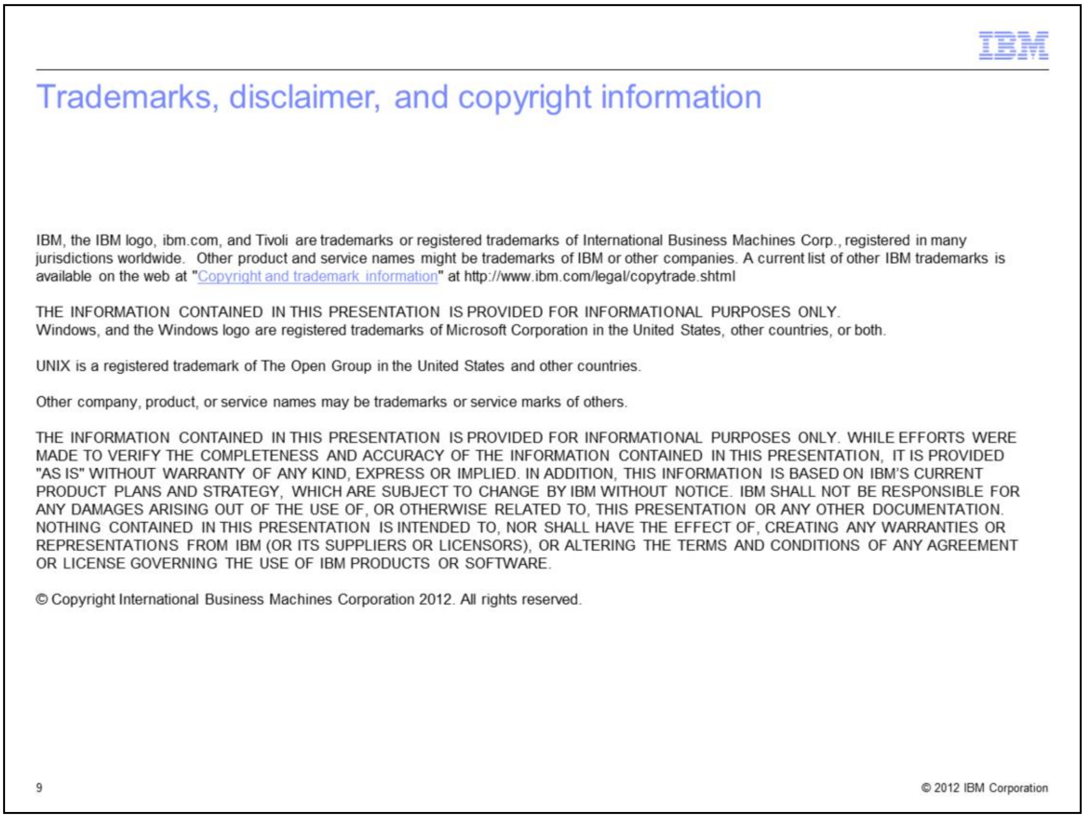
Basic elements

```
<tepgwml:gateway
xmlns:tepgwml="http://xml.schemas.ibm.com/tivoli/tep/kde/">
   <zone>
               <interface>        upstream interface
                              <bind>
                                             <connection>
                                             </connection>
                              </bind>
                              <interface>        downstream interface (embedded in upstream)
                                             <bind>
                                                            <connection>
                                                            </connection
                                             </bind>
                              </interface>
               </interface>
   </zone>
   <portpool>
   </portpool>
</tepgwml:gateway>
```

8 IBM Tivoli Monitoring 6.1 firewall implementation: KDE Gateway component © 2012 IBM Corporation

This is a basic structure for an XML file. When you define a zone, you will start defining the upstream interface.

First, you will define the connection, then within the upstream interface, you will define the downstream interface; so it’s embedded in the upstream interface. Then define some port pools. This is the basic structure.



Trademarks, disclaimer, and copyright information

IBM, the IBM logo, ibm.com, and Tivoli are trademarks or registered trademarks of International Business Machines Corp., registered in many jurisdictions worldwide. Other product and service names might be trademarks of IBM or other companies. A current list of other IBM trademarks is available on the web at "Copyright and trademark information" at http://www.ibm.com/legal/copytrade.shtml

THE INFORMATION CONTAINED IN THIS PRESENTATION IS PROVIDED FOR INFORMATIONAL PURPOSES ONLY.
Windows, and the Windows logo are registered trademarks of Microsoft Corporation in the United States, other countries, or both.

UNIX is a registered trademark of The Open Group in the United States and other countries.

Other company, product, or service names may be trademarks or service marks of others.

THE INFORMATION CONTAINED IN THIS PRESENTATION IS PROVIDED FOR INFORMATIONAL PURPOSES ONLY. WHILE EFFORTS WERE MADE TO VERIFY THE COMPLETENESS AND ACCURACY OF THE INFORMATION CONTAINED IN THIS PRESENTATION, IT IS PROVIDED "AS IS" WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED. IN ADDITION, THIS INFORMATION IS BASED ON IBM'S CURRENT PRODUCT PLANS AND STRATEGY, WHICH ARE SUBJECT TO CHANGE BY IBM WITHOUT NOTICE. IBM SHALL NOT BE RESPONSIBLE FOR ANY DAMAGES ARISING OUT OF THE USE OF, OR OTHERWISE RELATED TO, THIS PRESENTATION OR ANY OTHER DOCUMENTATION. NOTHING CONTAINED IN THIS PRESENTATION IS INTENDED TO, NOR SHALL HAVE THE EFFECT OF, CREATING ANY WARRANTIES OR REPRESENTATIONS FROM IBM (OR ITS SUPPLIERS OR LICENSORS), OR ALTERING THE TERMS AND CONDITIONS OF ANY AGREEMENT OR LICENSE GOVERNING THE USE OF IBM PRODUCTS OR SOFTWARE.