WebSphere Application Server architecture – federated

Unit objectives

After completing this unit, you should be able to:

- Describe the Network Deployment runtime flow

- Describe Network Deployment concepts and terminology, such as *cell*, *node*, *node agent*, and *deployment manager*

- Describe the Network Deployment administration flow

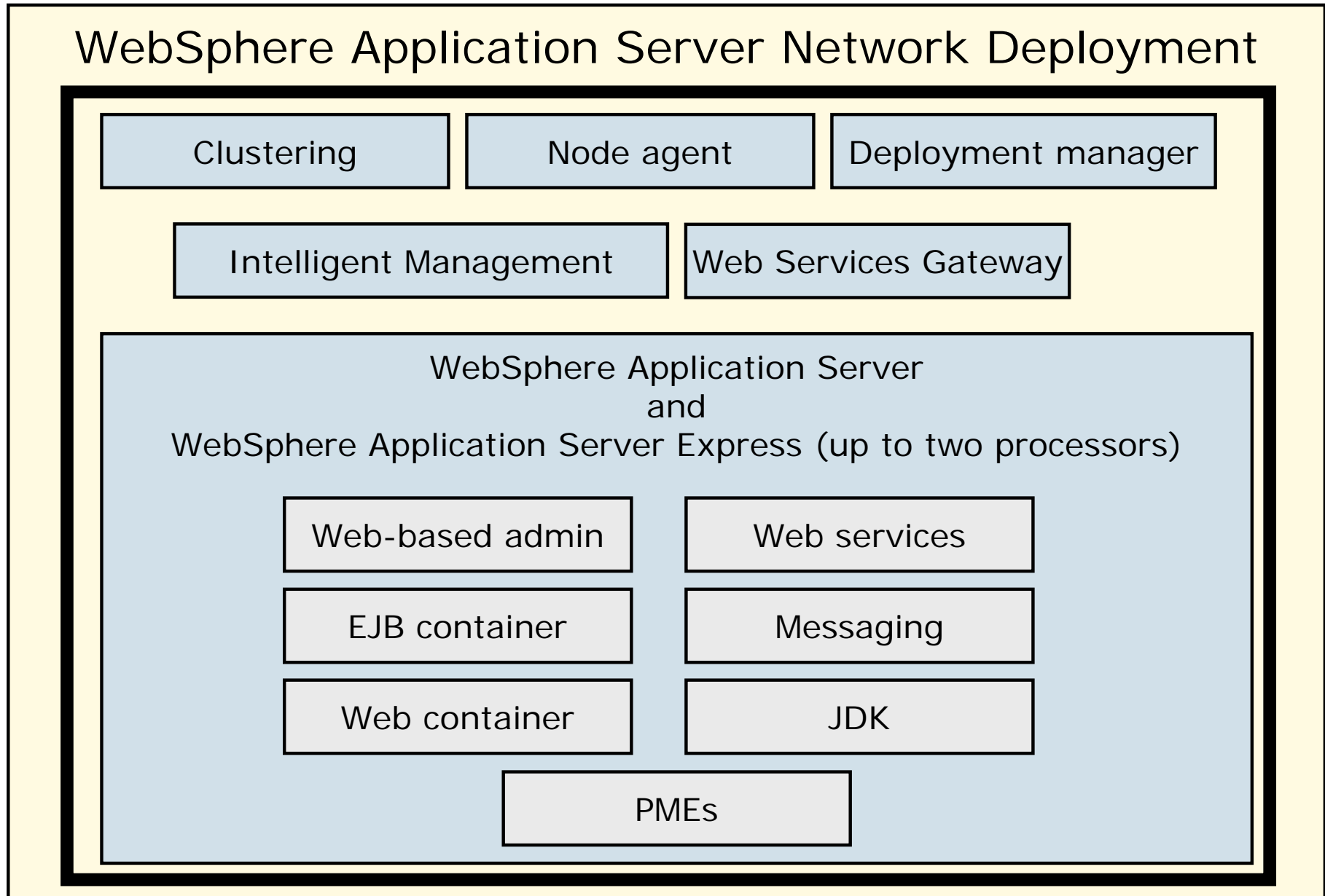- Explain how to manage web servers from WebSphere Application Server

Topics

- Network deployment concepts
- Managing web servers
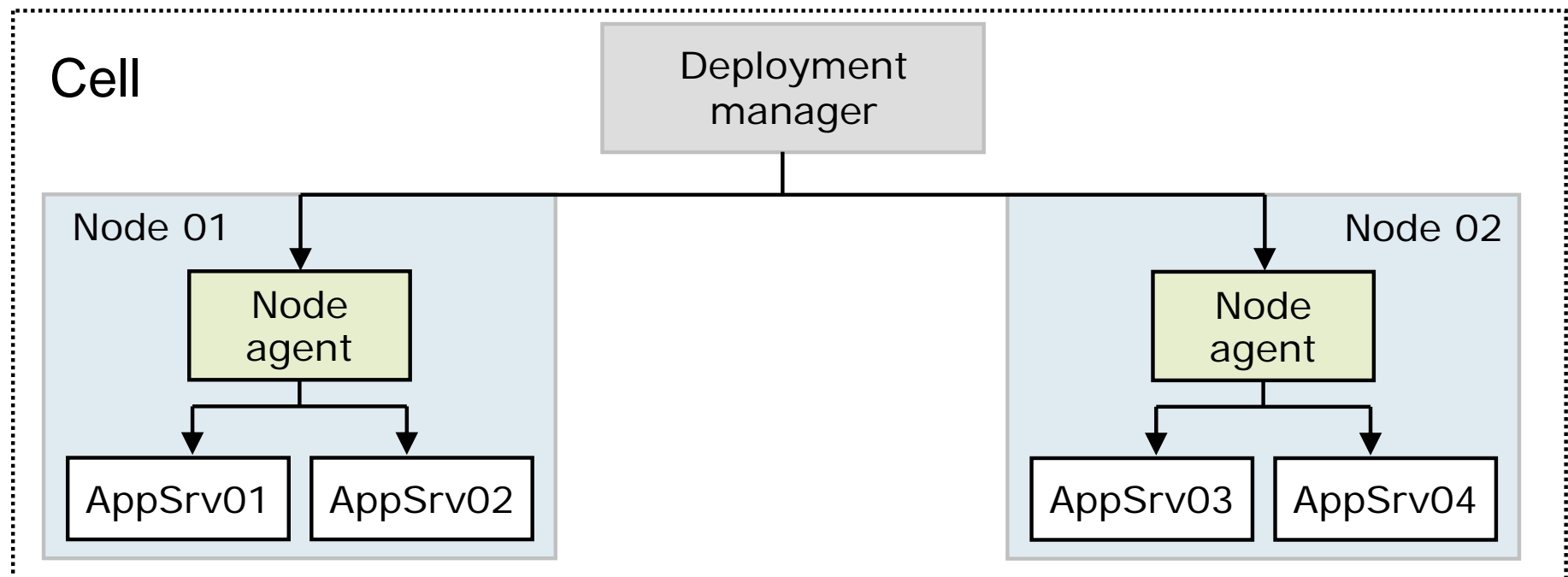- Additional concepts

Network deployment concepts

8.0

Version 8.5 packaging

WebSphere Application Server Network Deployment

Clustering	Node agent	Deployment manager

Intelligent Management	Web Services Gateway

**WebSphere Application Server
and
WebSphere Application Server Express (up to two processors)**

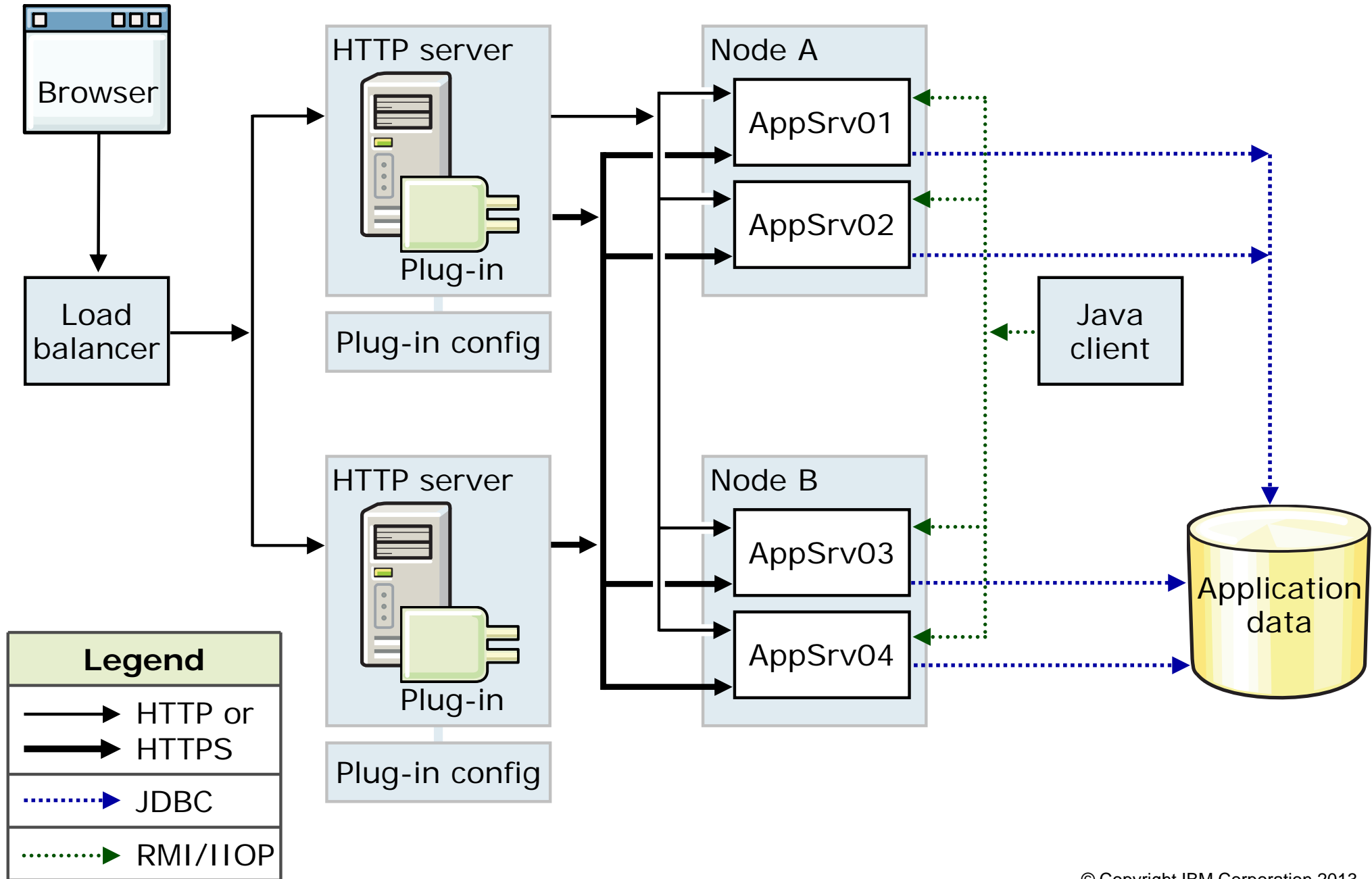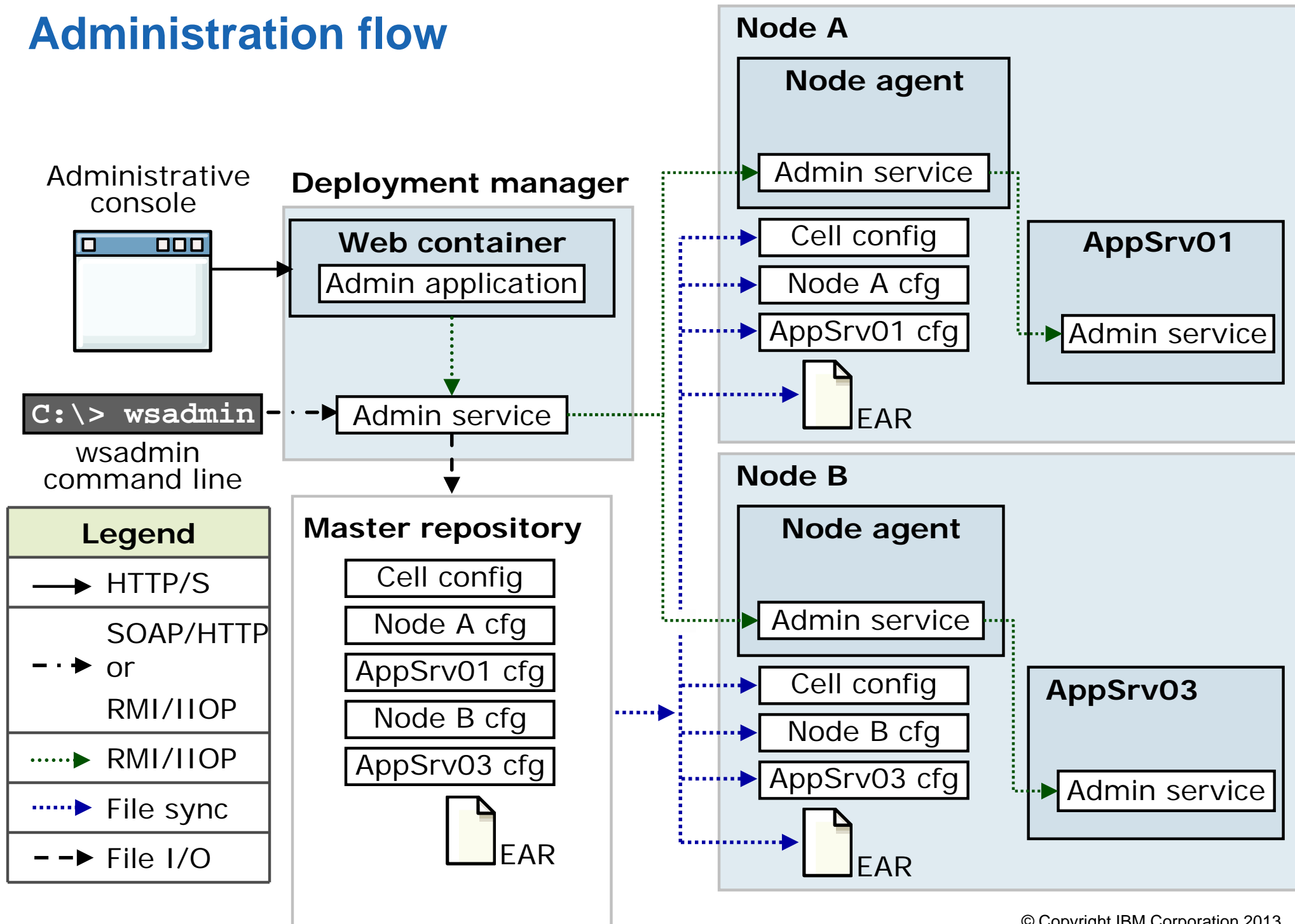|	
Web-based admin	Web services
EJB container	Messaging
Web container	JDK

PMEs

Network deployment concepts

- A **deployment manager** (DMgr) process manages the node agents
  - Holds the configuration repository for the entire management domain, called a **cell**
  - Within a cell, the administrative console runs inside the DMgr
- A **node** is a logical grouping of application servers
  - A single **node agent** process manages each node
  - Multiple nodes can exist on a single machine by using profiles

Cell

Deployment manager

Node 01

Node agent

AppSrv01 AppSrv02

Node 02

Node agent

AppSrv03 AppSrv04

Network deployment runtime flow

Browser

Load balancer

HTTP server

Plug-in

Plug-in config

Node A

AppSrv01

AppSrv02

Node B

AppSrv03

AppSrv04

Java client

Application data

Legend

→ HTTP or HTTPS

⇢ JDBC

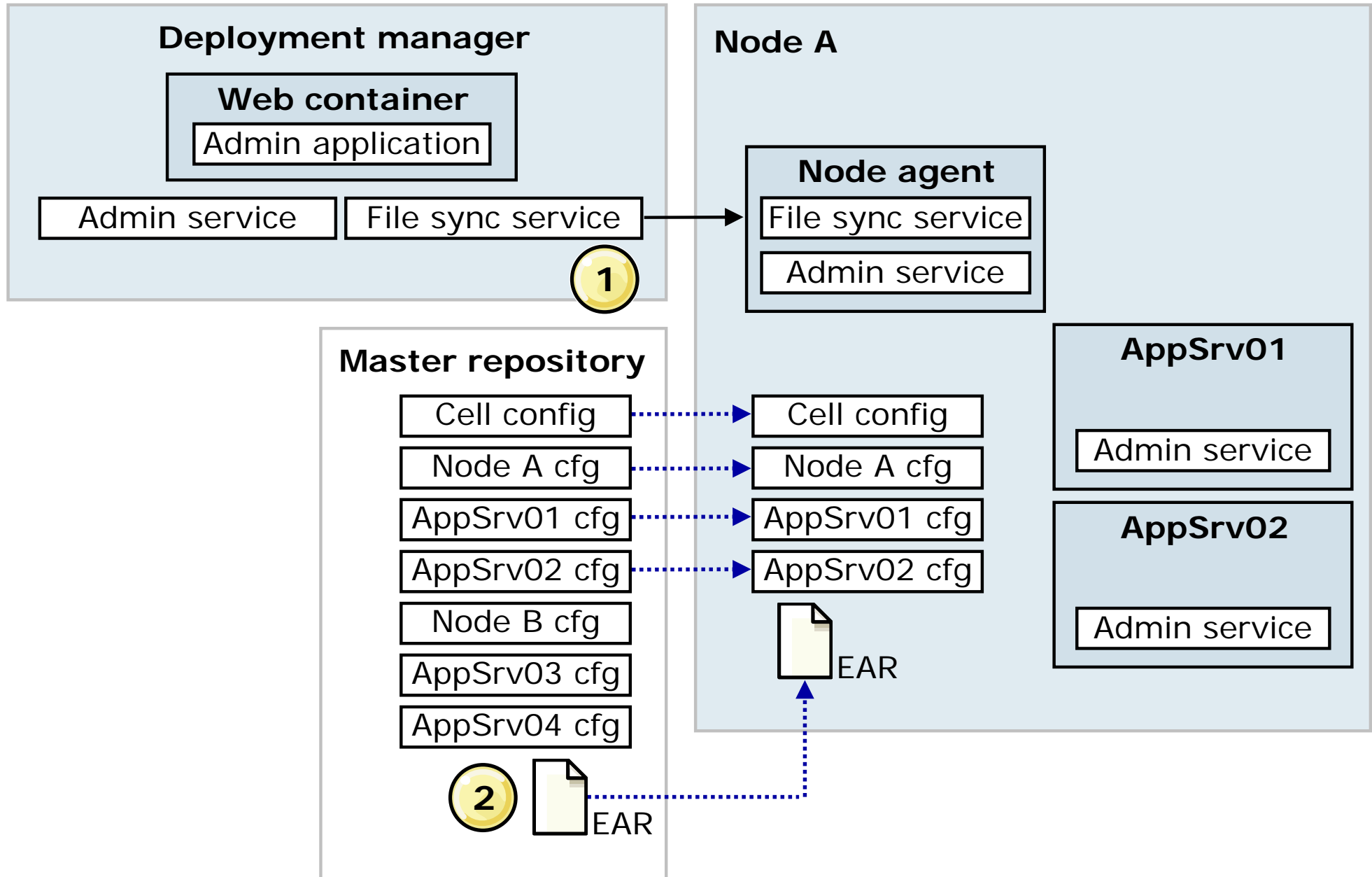⇢ RMI/IIOP

File synchronization

WebSphere Network Deployment profiles

Benefits of profiles in network deployment

- Think of profiles as representing a node
- Can install multiple profiles on a single machine

All profiles use the same product files

- **Application server** profile (stand-alone)
    - Equivalent to Base or Express application server
    - Has a node name and a cell name property, and corresponding directories
    - Cell directory is overwritten upon federation
- **Deployment manager** profile
    - Creates a deployment manager
- **Custom** profile (managed)
    - Creates a managed node which, by default, is federated into a cell
    - Creates a node agent, but no application servers
- **Cell** profile
    - Creates both a deployment manager and a federated node
- Others

Managing web servers

8.0

Web servers within a cell

Web servers are customized

- Each web server plug-in is customizable
- Requires a web server definition
    - Defining a web server does not mean that it is managed
- Plug-in properties are defined on a per web server basis
- Each plug-in has a unique `plugin-cfg.xml` generated for it
- A cell level `plugin-cfg.xml` can also be generated

Web servers can optionally be managed

- Web servers can be unmanaged
    - No management is available
- You can manage web servers by:
    - A node agent
    - The IBM HTTP Server administrative process

Managed versus unmanaged nodes

- A managed node is a node that contains a node agent

- An unmanaged node is a node in the cell without a node agent
  - The rest of the environment can be aware of the node
  - Useful for defining HTTP servers as part of the topology
  - Allows creation of different plug-in configurations for different HTTP servers

Cell

Deployment manager

Unmanaged node

Node 02

Node agent

AppSrv03   AppSrv04
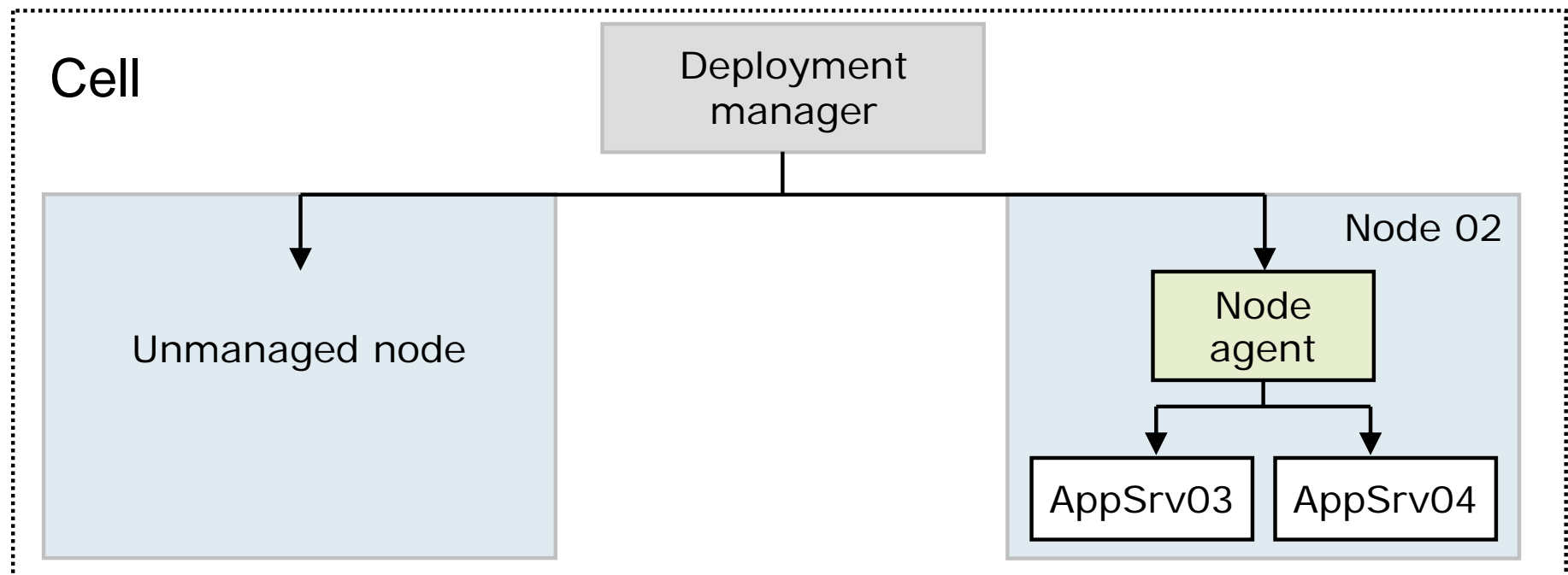
Web server definitions (1 of 2)

- Web server definitions are created to allow the mapping of Java EE enterprise applications to specific web servers
  - Can be done through the administrative console
  - By default, all currently installed applications are mapped to a web server created using the admin console

Specify a Web server name, type, host name and platform.

* Server name
webserver1

* Type
IBM HTTP Server

* Host name

was85host

* Platform
Linux

Enter the Web server properties.

✱ Port
80

✱ Web server installation location
/opt/IBM/HTTPServer

✱ Plug-in installation location
/opt/IBM/HTTPServer/Plugins
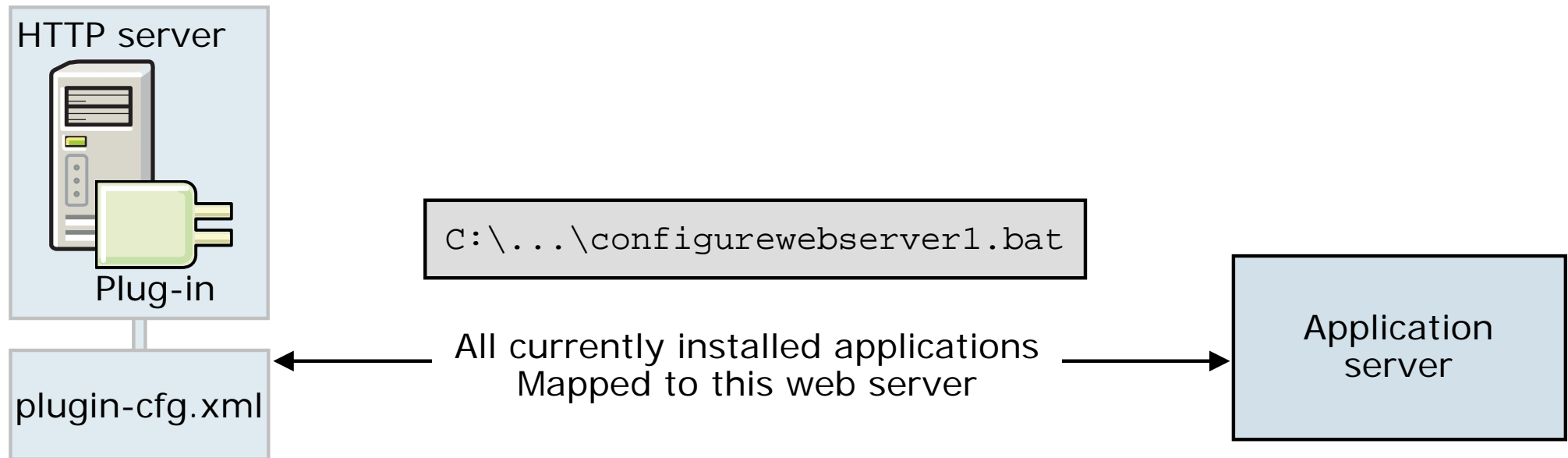
Application mapping to the Web server
All

Web server definitions (2 of 2)

- Alternately use the script that is generated during the configuration of the plug-in, which can automate the mapping of all the applications to the web server

    `configure<Web_server_name>.bat` in `<plugin_root>\bin`

HTTP server

Plug-in

plugin-cfg.xml

`C:\...\configurewebserver1.bat`

All currently installed applications
Mapped to this web server

Application server

Web server custom plugin-cfg.xml

- Mapping the applications to specific web servers causes the `plugin-cfg.xml` files for only those web servers to include the information for those applications
  - Target-specific web server applications that run in a cell
  - Deployment manager automatically generates them

Application module explicitly mapped to a web server

Managing plugin-cfg.xml files

- `plugin-cfg.xml` files are automatically generated and propagated
- This behavior is the default, but can be changed
- This behavior is configurable through the console

`plugin-cfg.xml` files can be generic to a cell or custom to web server

- Generating a cell generic `plugin-cfg.xml` file
  - Use the command-line script *<was_root>*\bin\GenPluginCfg.bat
  - Not available through the console
- Generating a web server custom `plugin-cfg.xml` file
  - Use the administrative console
  - Must map applications to web servers
  - Can customize the plug-in settings of each web server

Web servers

Web servers

Use this page to view a list of the installed Web servers

⊞ Preferences

Generate Plug-in	Propagate Plug-in	New	Delete	Templates...	Start	Stop	Terminate

Select	Name ⌃	Web server Type ⌃	Node ⌃	Host Name ⌃	Version ⌃	Status ↻
	You can administer the following resources:					
☑	webserver01	IBM HTTP Server	ihsnode	was7host01	Not applicable	➡

Total 1

Managing web server plug-in properties

Plug-in properties

☐ Ignore DNS failures during Web server startup

Additional Properties

- Request and Response
- Caching
- Request Routing
- Custom Properties

* Refresh configuration interval

`60` seconds

Repository copy of Web server plug-in files:

* Plug-in configuration file name

`plugin-cfg.xml` [View]

☑ Automatically generate the plug-in configuration file

☑ Automatically propagate plug-in configuration file

* Plug-in key store file name

`plugin-key.kdb`

[Manage keys and certificates]

[Copy to Web server key store directory]

Web server copy of Web server plug-in files:

* Plug-in configuration directory and file name

`c:\Program Files\IBM\HTTPServer\Plugins\config\webserver01\plugin-cfg.x`

* Plug-in key store directory and file name

`c:\Program Files\IBM\HTTPServer\Plugins\config\webserver01\plugin-key.`

- Each web server can have customized plug-in configuration settings
  - Not just application mappings

Virtual hosts

- Configuration that allows a host machine to resemble multiple host machines

- Each virtual host has a logical name and one or more host aliases

- There are several default virtual hosts, including:

  - default_host: used for accessing the default applications
    Example: `http://localhost:9080/snoop`

  - admin_host: used for accessing the administrative console
    Example: `http://localhost:9060/ibm/console`

Browser

HTTP server

Plug-in

Application server

Virtual host important here
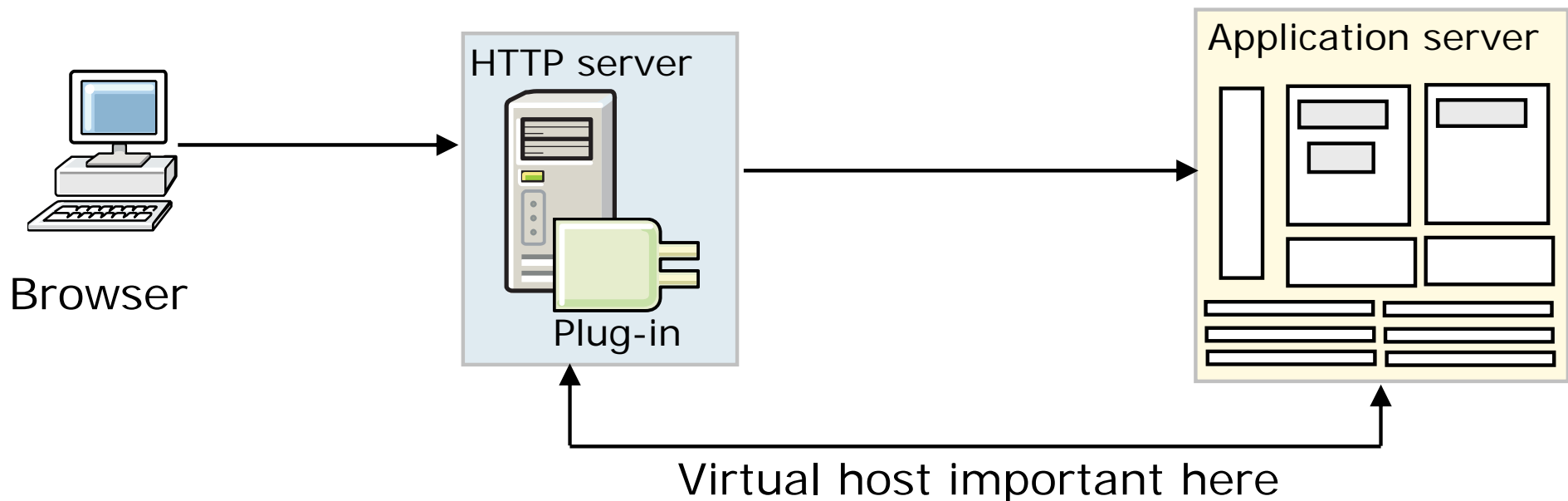
Defining virtual hosts

New   Delete

Select   Name

☐   admin_host

☐   default_host

☐   proxy_host

Total 3

Virtual Hosts

Virtual Hosts > default_host

Use this page to create a virtual host with a unique set of Web access ports. Such a configuration lets a single host machine resemble multiple host machines. Each virtual host has a logical name and a list of one or more domain name system (DNS) aliases by which it is known.

Configuration

General Properties

✱ Name

default_host

Additional Properties

▪ Host Aliases

▪ MIME Types

Apply   OK   Reset   Cancel

Virtual Hosts

Virtual Hosts > default_host > Host Aliases

Use this page to edit, create, or delete a domain name system (DNS) alias by which the virtual host is known.

⊞ Preferences

New   Delete

Select	Host Name	Port
☐	*	9080
☐	*	80
☐	*	9443
☐	*	5060
☐	*	5061
☐	*	443

Total 6

Managing web servers with WebSphere

Deployment manager can manage external web servers

- IBM HTTP Server (special case; no node agent needed)
  - Deployment manager can distribute `plugin-cfg.xml` files to web server machines
  - Can be started and stopped
  - Can edit the `httpd.conf`
- Other web servers (node agent needed)
  - Can have `plugin-cfg.xml` files that are automatically distributed to them
  - Can be started and stopped

Web servers are defined within WebSphere cell topologies

- Managed node (local) or unmanaged node (remote)
  - Managed nodes use a node agent to control the web server
  - Unmanaged nodes use the IBM HTTP Server Admin Service instead of a node agent to control the web server

Unmanaged web server
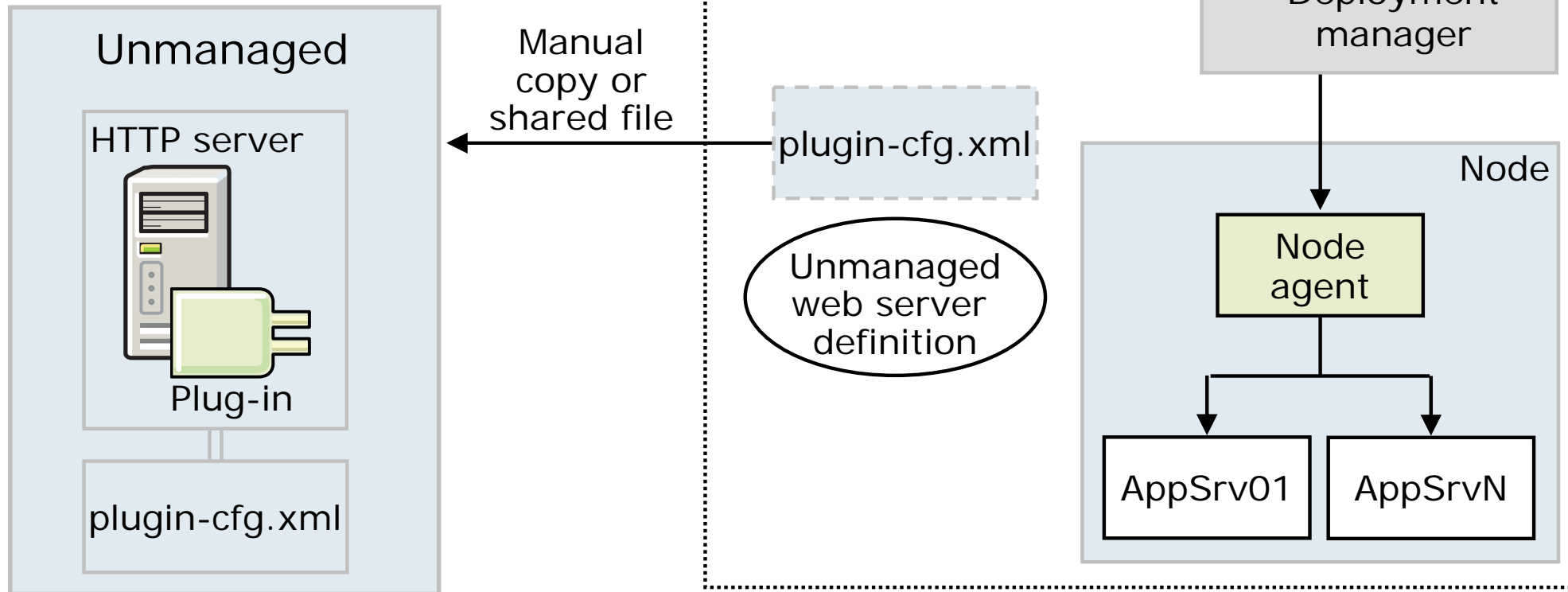
- WebSphere node agent does not manage web server (other than IBM HTTP Server)
  - Allows WebSphere system administrator to create custom plug-in files for a specific web server
  - Application mappings
  - SSL certificates
- Manually copy or use FTP to transfer the plug-in configuration file from the deployment manager machine to the web server machine

Managed web server on a managed node (local)
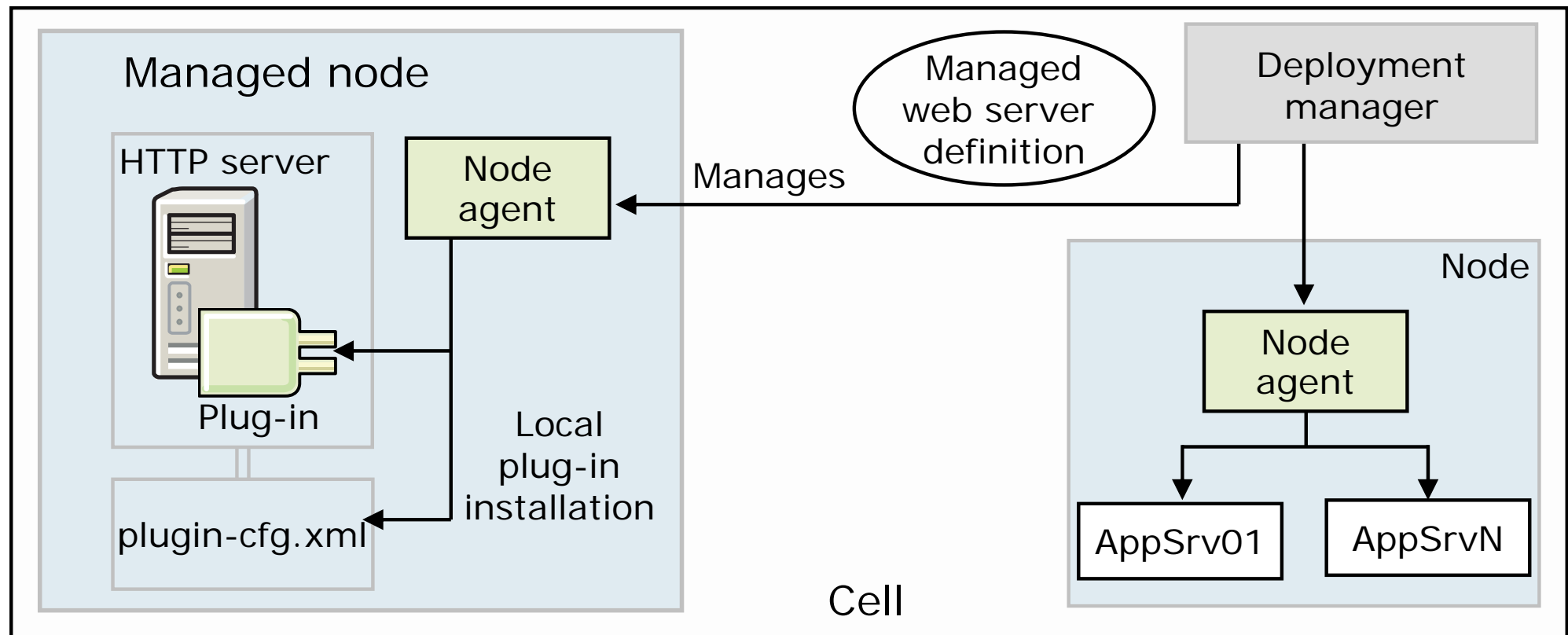
- Install a web server on a managed node
- Create a web server definition within the DMgr
- Node agent receives commands from DMgr to administer the web server
- `plugin-cfg.xml` file is propagated through the file synchronization service and is in the `config` directory
- Warning: security issues if this configuration spans a DMZ

IBM HTTP Server as unmanaged node (remote)
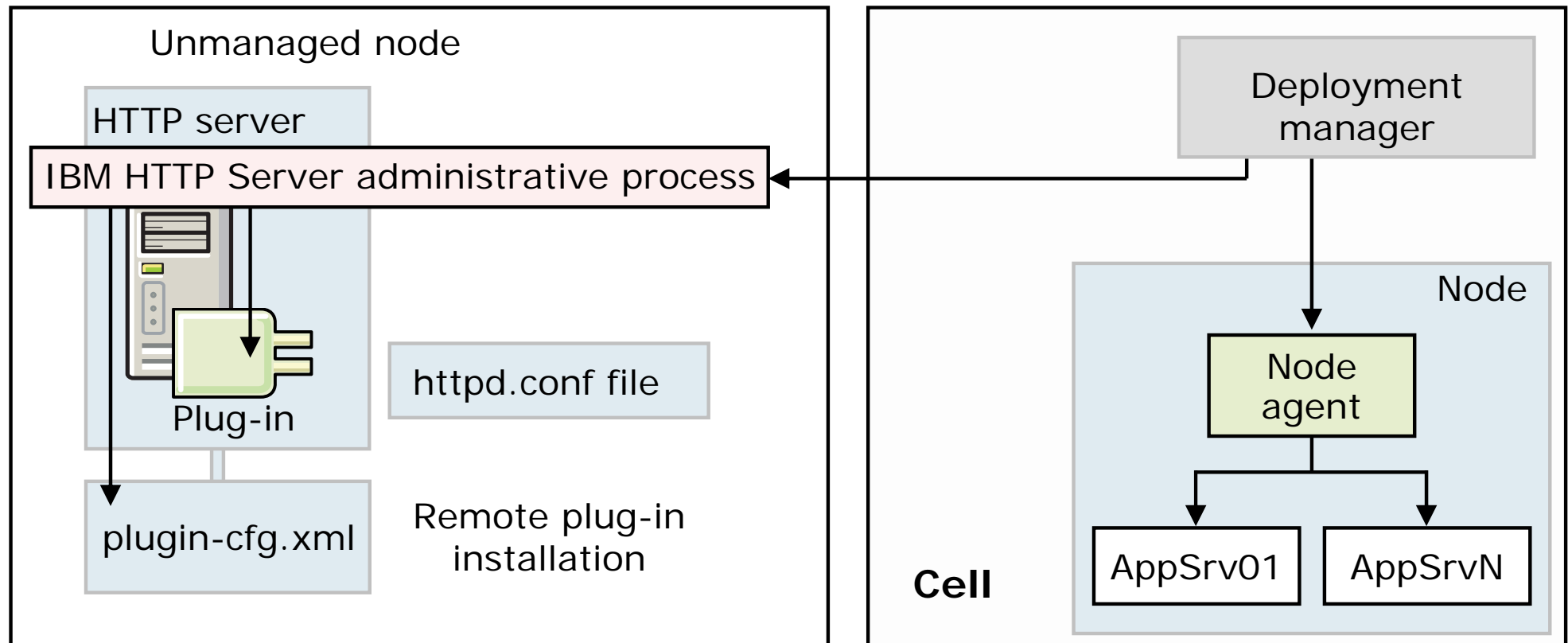
- The IBM HTTP Server administrative process provides administrative functions for IBM HTTP Server within WebSphere
  - Able to start, stop IBM HTTP Server, make configuration changes to `httpd.conf`, and automatically push the plug-in configuration file to IBM HTTP Server machine
  - Node agent is not needed on the web server machine

IBM HTTP Server administration overview

- Direct administration of IBM HTTP Server V7 by manually editing `httpd.conf`
- There is no web-based console for IBM HTTP Server as there was in previous versions
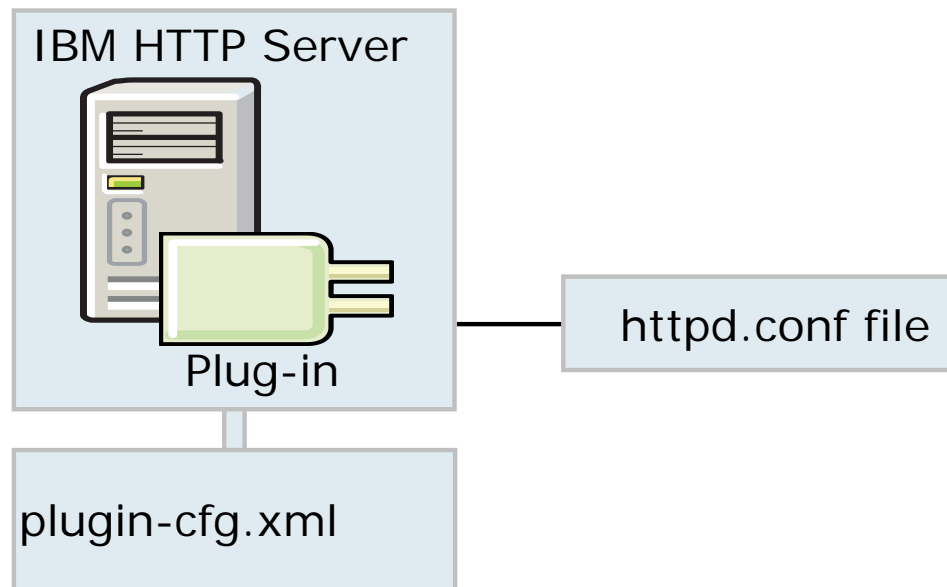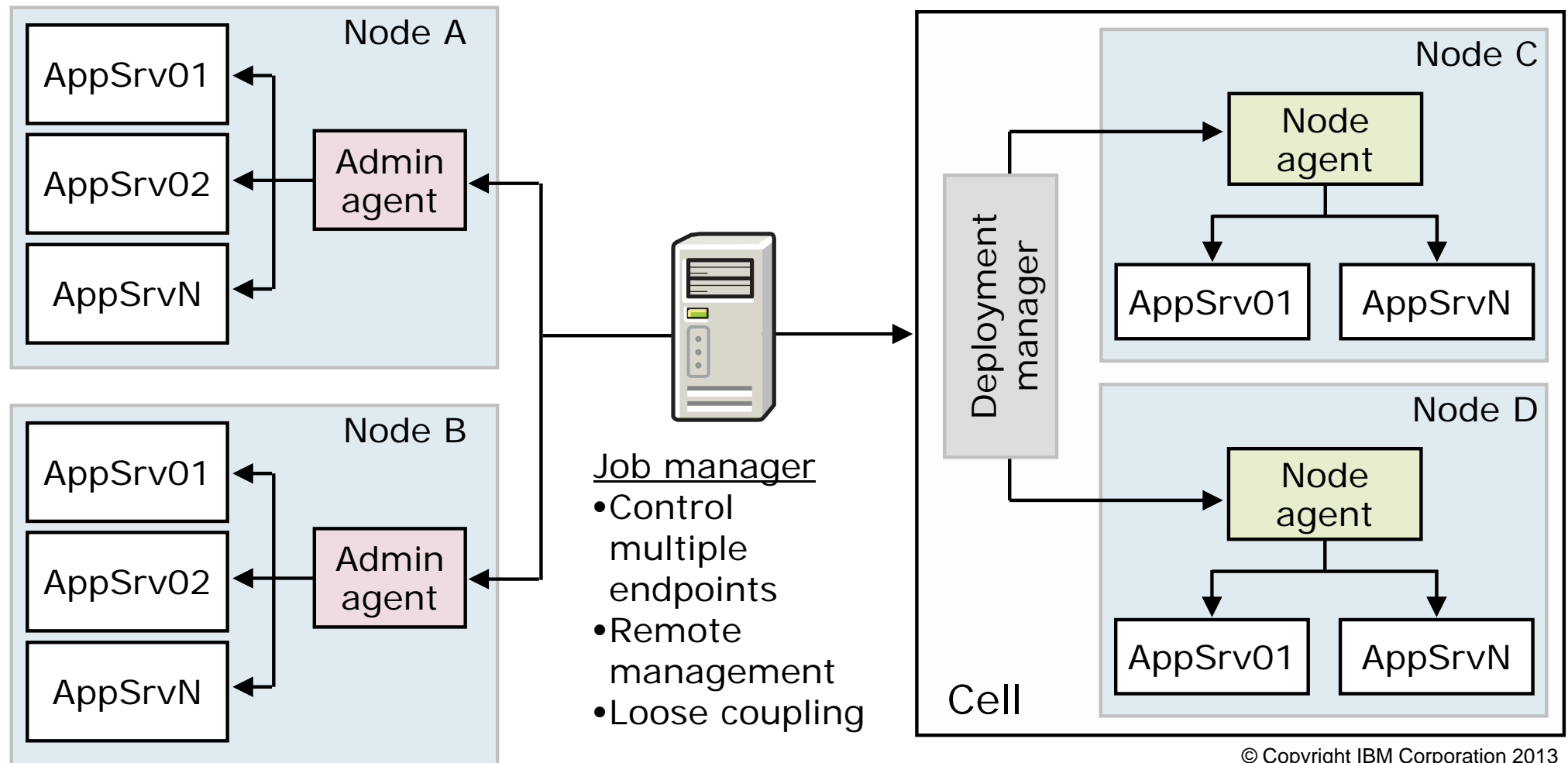
IBM HTTP Server administration server

- IBM HTTP Server administration server runs as a separate instance of IBM HTTP Server

- Administrative component for IBM HTTP Server V7 includes:
  - IBM HTTP Server administration configuration file (`admin.conf`)
  - Default port for the IBM HTTP Server administration server is 8008

- IBM HTTP Server administration authentication password file (`admin.passwd`)
  - Initially blank, which prohibits access to IBM HTTP Server administration
  - Administrator updates IBM HTTP Server admin password file by using:
    ```
    > htpasswd -cm ..\conf\admin.passwd <user_name>
    ```

- To start and stop the administration server:
  - `<ihs_root>/bin/adminctl start`
  - `<ihs_root>/bin/adminctl stop`
  - Or Windows service

Additional concepts

© Copyright IBM Corporation 2013
Course materials may not be reproduced in whole or in part without the prior written permission of IBM.

8.0

Flexible management

- Loose management coupling

- Coordinates management across a group of endpoints

  – One job to install application across a number of nodes

- Can manage through administrative agent or deployment manager

Node A

AppSrv01

AppSrv02

AppSrvN

Admin agent

Node B

AppSrv01

AppSrv02

AppSrvN

Admin agent

Job manager
- Control multiple endpoints
- Remote management
- Loose coupling

Deployment manager

Node C

Node agent

AppSrv01   AppSrvN

Node D

Node agent

AppSrv01   AppSrvN

Cell

Centralized Installation Manager (CIM)

- Simplifies the installation and maintenance of application servers within a Network Deployment cell

- Install, update, uninstall version 8.x and all Installation Manager installable products

- For V7 nodes in a v8.x cell
  - Pushes remote binary files or maintenance to remote targets
  - Starts the standard or update installer to complete the installation of the update

Intelligent runtime provisioning

- Dynamic start of server components that are based on application needs
- Reduces runtime footprint; less memory required
- Can significantly reduce startup times
- Disabled by default

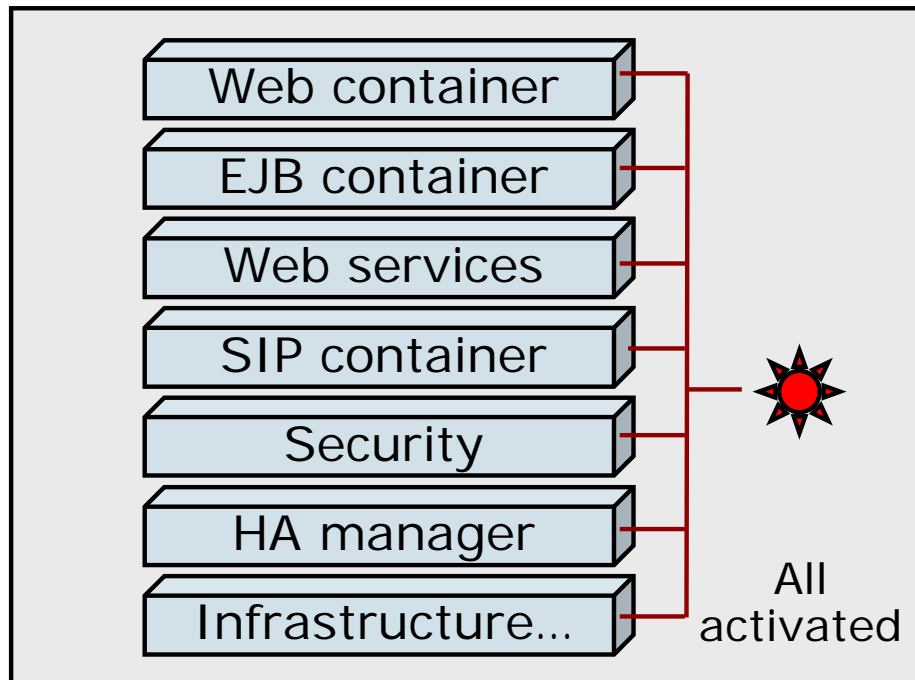**General Properties**

Name

server1
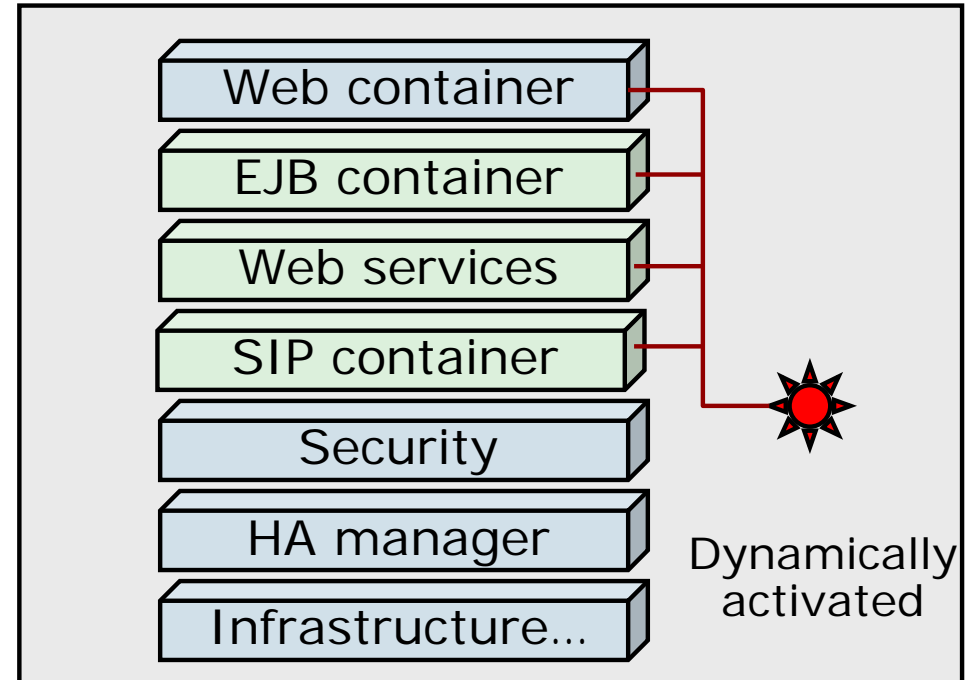
Node name

was85hostNode01

☐ Run in development mode

☑ Parallel start

☑ Start components as needed

V6.1 server

| Web container |
| EJB container |
| Web services |
| SIP container |
| Security |
| HA manager |
| Infrastructure... |

All activated

V8.5 server

| Web container |
| EJB container |
| Web services |
| SIP container |
| Security |
| HA manager |
| Infrastructure... |

Dynamically activated

© Copyright IBM Corporation 2013

Edge Components

- WebSphere Application Server Network Deployment package contains the following Edge Components functions:
  - Load balancer
  - Caching proxy

Client   Caching proxy   Load balancer   Cluster of load-balanced servers

- Edge Components install separately from WebSphere Application Server

- Load balancer is responsible for balancing the load across multiple servers that can be within either local area networks or wide area networks

- Purpose of caching proxy is to reduce network congestion within an enterprise by offloading security and content delivery from web servers and application servers
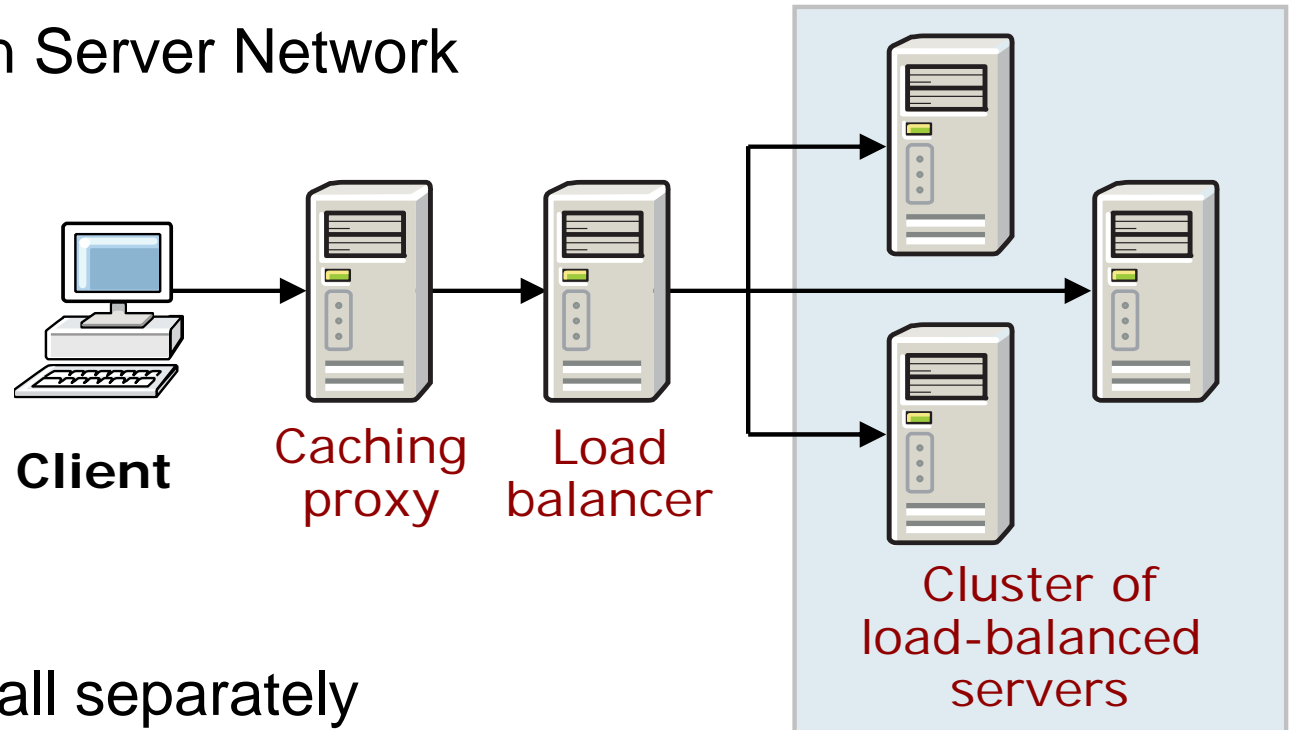
Intelligent Management

Health management	Application edition management	Performance management
Self protecting Self healing	Self managing	Self optimizing

App version 1.0 | App version 2.0

Intelligent routing
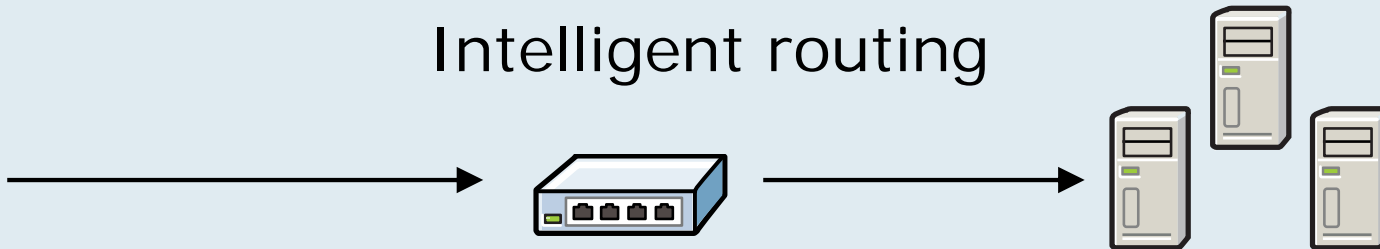
Unit summary

Having completed this unit, you should be able to:

- Describe the Network Deployment runtime flow
- Describe Network Deployment concepts and terminology, such as *cell, node, node agent,* and *deployment manager*
- Describe the Network Deployment administration flow
- Explain how to manage web servers from WebSphere Application Server

Checkpoint questions

1. A process that handles communications with the resources within the node is _____.

2. What is the process when the node agent checks for changes to the master configuration?

3. What is a configuration that allows a host machine to resemble multiple host machines?

4. What defines the runtime environment for either the deployment manager or the application server?

Checkpoint answers

1. A process that handles communications with the resources within the node is the _____.

   Node agent

2. What is the process when the node agent checks for changes to the master configuration?

   File synchronization

3. What is a configuration that allows a host machine to resemble multiple host machines?

   Virtual host

4. What defines the runtime environment for either the deployment manager or the application server?

   Profiles