LABWORK 1

XML Web Services design

Tasklist.

1) Organize portal for provision of modern Xml Web Services.
2) Define four network segments for this portal:
    i. Web farm segment for support of web services.
    ii. Datacenter segment for content storage and deliver.
    iii. Corporate users segment for support of inner users.
    iv. Wireless segment for mobility of inner users.
3) Join all segments using switch distribution system (VLAN technology); use hierarchical network structure.
4) Attach portal to the Internet using high speed router.
5) Add three DNS servers: one outer server for name resolving for Internet users, one inner server for name resolving for whole portal and one local server for corporate users segment.
6) Add certificate server for credential management for Internet users.
7) Add second Internet connection of datacenter via designated router and switch.
8) Define data center network:
    i. Define four network segments: 1 – file server cluster, 2 – database server cluster, 3 – content servers of different types, 4 – data center management network.
    ii. Add switches (with high-speed uplinks) for collaboration with corporate users segment, designated Internet router and web server farm.
    iii. Join all network segments using switch system (mesh topology) with early-added switches.
    iv. Locate real-time content management server and several file servers in first segment.
    v. Locate several database servers in second segment.
    vi. Locate streaming media server and ftp server in third segment.
    vii. Locate content management server and content directory server in fourth segment.

viii. Join all servers of fourth segment with corporate user's segment switch and web server farm switch.

9) Define web farm segment:
   i. Add four web servers and join them using load balancing hub.
   ii. Add three switches and five application servers.
   iii. Join web servers and switches using mesh topology network.
   iv. Join application servers and switches using mesh topology network.
   v. Add datacenter collaboration switch (with high-speed uplink to data center web farm collaboration switch).
   vi. Join each application server with datacenter collaboration switch.
   vii. Add control server for web farm management. Join it with all switches.

10) Define corporate user's segment:
   i. Define several domains for different groups of users.
   ii. Add primary controller for directory. Attach it to corporate user's switch.
   iii. Add one secondary controller for each domain.
   iv. Add several workstations for each domain.
   v. Add application server, database server and corporate web server to special domain.

11) Define wireless segment as single access point (wireless segment is not point of interest for this lab).

Additional useful information.

XML Web services are the fundamental building blocks in the move to distributed computing on the Internet. Open standards and the focus on communication and collaboration among people and applications have created an environment where XML Web services are becoming the platform for **application integration**. Applications are constructed using multiple XML Web services from various sources that **work together** regardless of where they reside or how they were implemented.

There are probably as many definitions of XML Web Service as there are companies building them, but almost all definitions have these things in common:

– XML Web Services expose useful functionality to Web users through a standard Web protocol. In most cases, the protocol used is **SOAP**.

- XML Web services provide a way to describe their interfaces in enough detail to allow a user to build a client application to talk to them. This description is usually provided in an XML document called a Web Services Description Language (**WSDL**) document.
- XML Web services are registered so that potential users can find them easily. This is done with Universal Discovery Description and Integration (**UDDI**).

SOAP

Soap is the communications protocol for XML Web services. SOAP is a specification that defines the XML format for messages – and that's about it for the required parts of the spec. If you have a well-formed XML fragment enclosed in a couple of SOAP elements, you have a SOAP message.

There are other parts of the SOAP specification that describe how to represent program data as XML and how to use SOAP to do Remote Procedure Calls. These optional parts of the specification are used to implement RPC-style applications where a SOAP message containing a callable function, and the parameters to pass to the function, is sent from the client, and the server returns a message with the results of the executed function. Most current implementations of SOAP support RPC applications because programmers who are used to doing COM or CORBA applications understand the RPC style. SOAP also supports document style applications where the SOAP message is just a wrapper around an XML document. Document-style SOAP applications are very flexible and many new XML Web services take advantage of this flexibility to build services that would be difficult to implement using RPC.

The last optional part of the SOAP specification defines what an HTTP message that contains a SOAP message looks like. This HTTP binding is important because HTTP is supported by almost all current OS's (and many not-so-current OS's). The HTTP binding is optional, but almost all SOAP implementations support it because it's the only standardized protocol for SOAP

WSDL

WSDL (often pronounced whiz-dull) stands for Web Services Description Language. For our purposes, we can say that a WSDL file is an XML document that describes a set of SOAP messages and how the messages are exchanged. In other

words, WSDL is to SOAP what IDL is to CORBA or COM. Since WSDL is XML, it is readable and editable but in most cases, it is generated and consumed by software.

To see the value of WSDL, imagine you want to start calling a SOAP method provided by one of your business partners. You could ask him for some sample SOAP messages and write your application to produce and consume messages that look like the samples, but this can be error-prone. For example, you might see a customer ID of 2837 and assume it's an integer when in fact it's a string. WSDL specifies what a request message must contain and what the response message will look like in unambiguous notation.

The notation that a WSDL file uses to describe message formats is based on the XML Schema standard which means it is both programming-language neutral and standards-based which makes it suitable for describing XML Web services interfaces that are accessible from a wide variety of platforms and programming languages. In addition to describing message contents, WSDL defines where the service is available and what communications protocol is used to talk to the service. This means that the WSDL file defines everything required to write a program to work with an XML Web service. There are several tools available to read a WSDL file and generate the code required to communicate with an XML Web service. Some of the most capable of these tools are in Microsoft Visual Studio® .NET.

UDDI

Universal Discovery Description and Integration is the yellow pages of Web services. As with traditional yellow pages, you can search for a company that offers the services you need, read about the service offered and contact someone for more information. You can, of course, offer a Web service without registering it in UDDI, just as you can open a business in your basement and rely on word-of-mouth advertising but if you want to reach a significant market, you need UDDI so your customers can find you.

A UDDI directory entry is an XML file that describes a business and the services it offers. There are three parts to an entry in the UDDI directory. The "white pages" describe the company offering the service: name, address, contacts, etc. The "yellow pages" include industrial categories based on standard taxonomies such as the North American Industry Classification System and the

Standard Industrial Classification. The "green pages" describe the interface to the service in enough detail for someone to write an application to use the Web service. The way services are defined is through a UDDI document called a Type Model or tModel. In many cases, the tModel contains a WSDL file that describes a SOAP interface to an XML Web service, but the tModel is flexible enough to describe almost any kind of service.

Visio software

Microsoft Visio is a 2D-object drawing application and is part of the Microsoft Office suite. The current version, Microsoft Visio 2010 for Windows, is available in three editions: Standard, Professional and Premium. The Standard and Professional editions both share the same interface, but the latter has additional templates for more advanced diagrams and layouts as well as unique functionality that intends to make it easy for users to connect their diagrams to a number of data sources and display the information graphically.

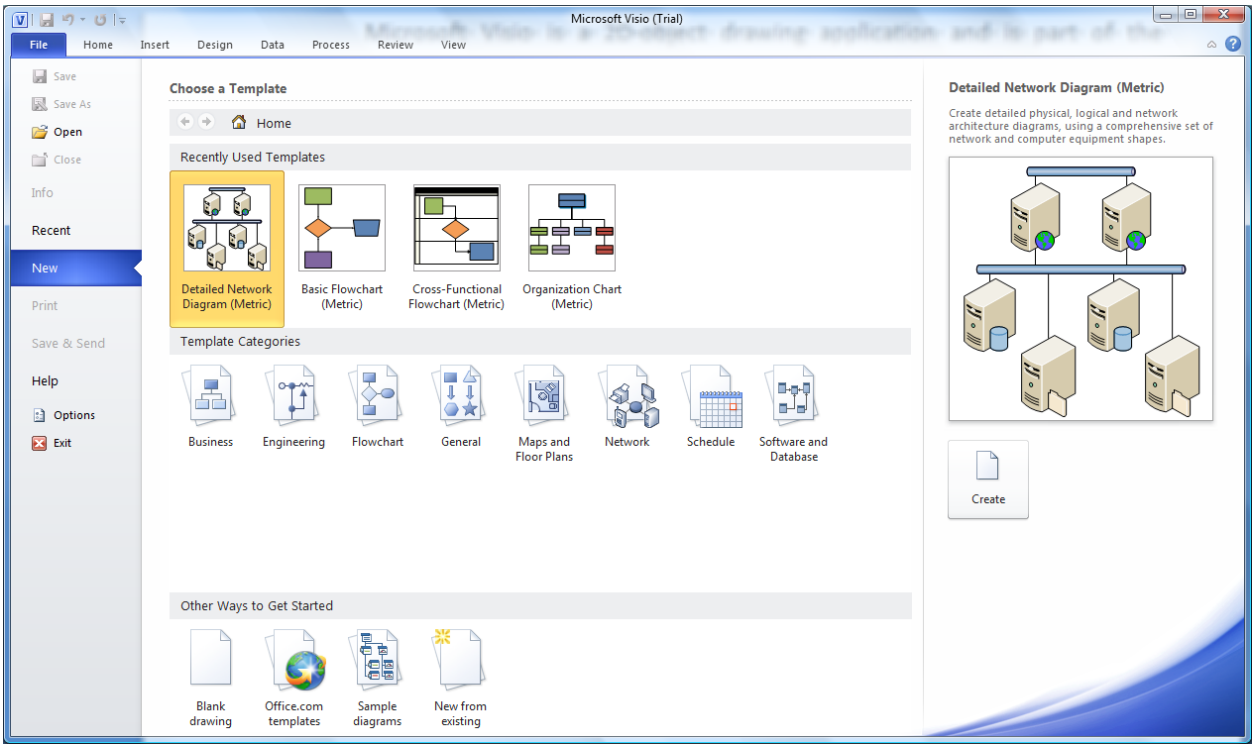
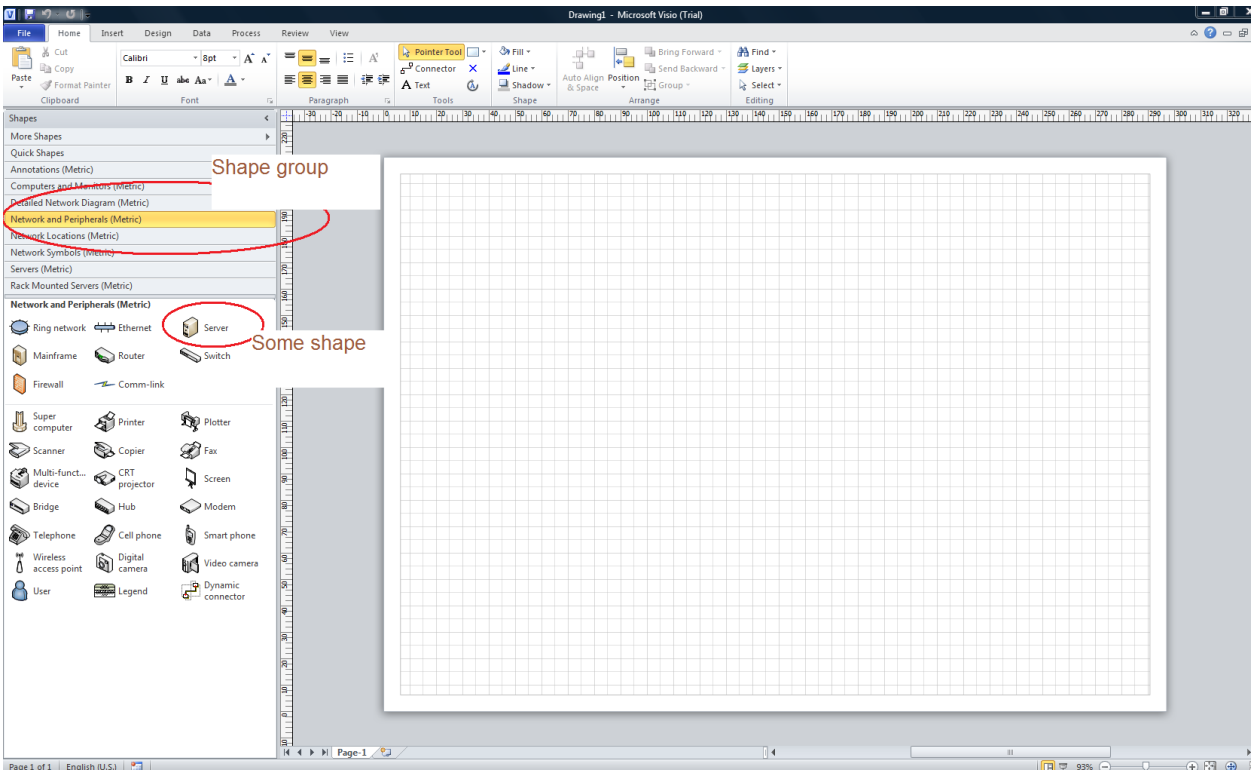
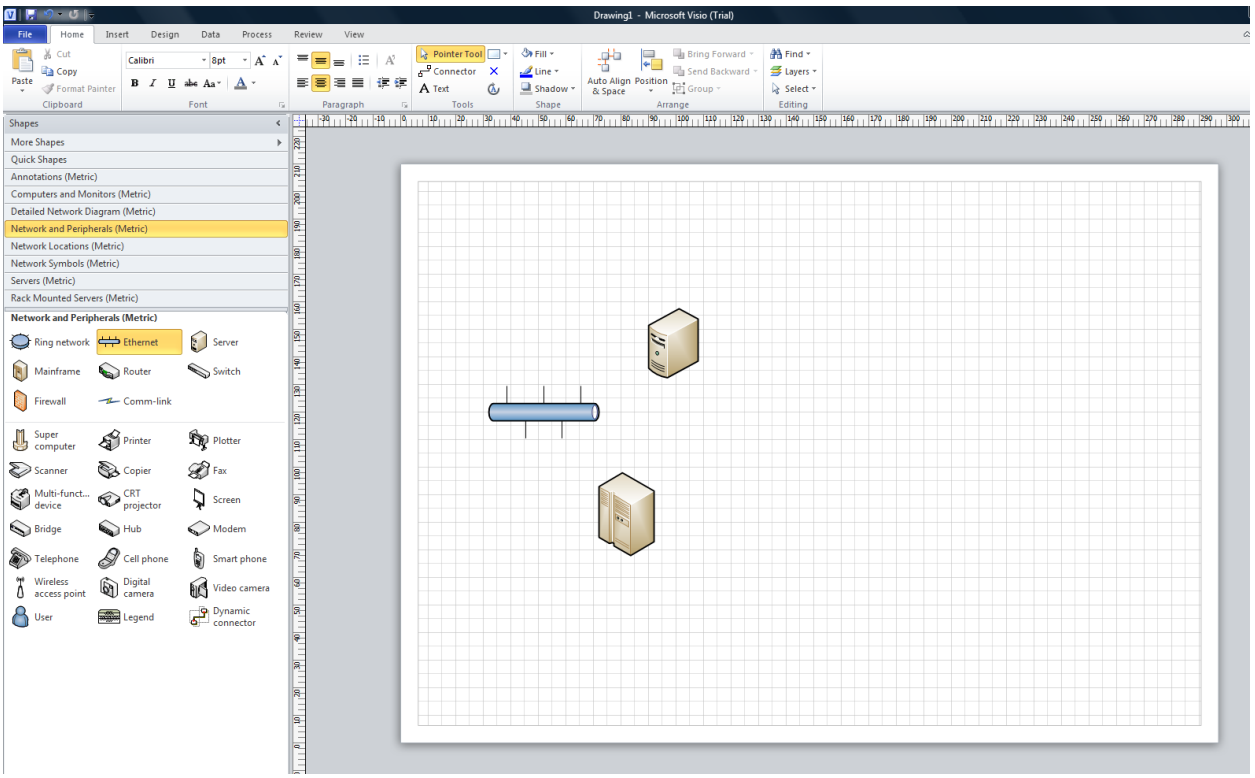


Figure – Visio initial page

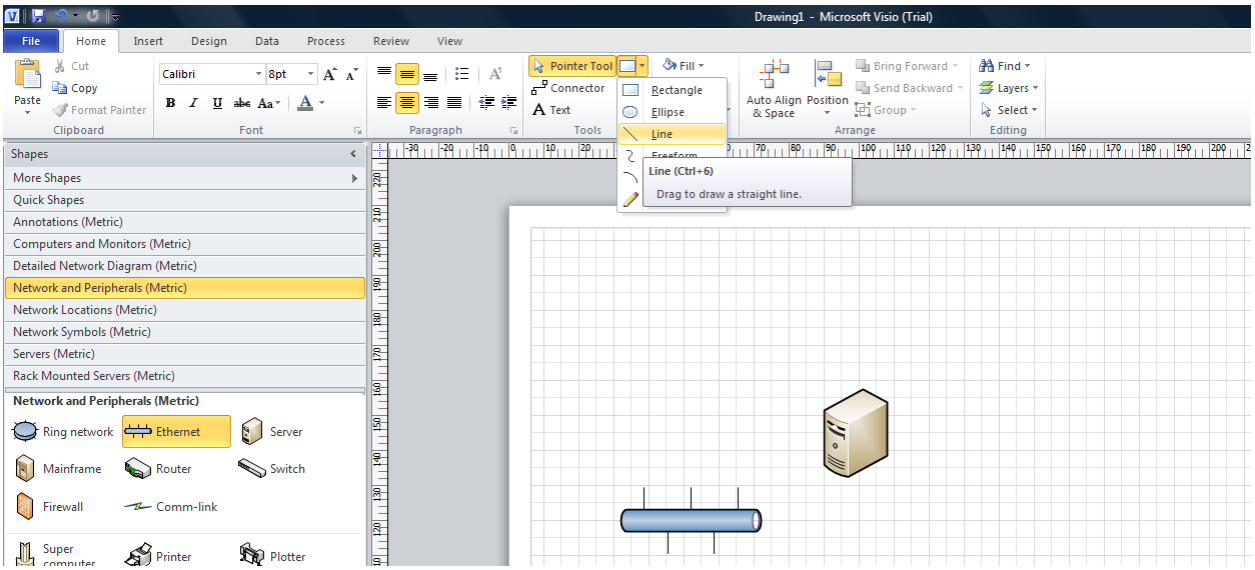
You can create network schemes using Detailed Network Diagram:



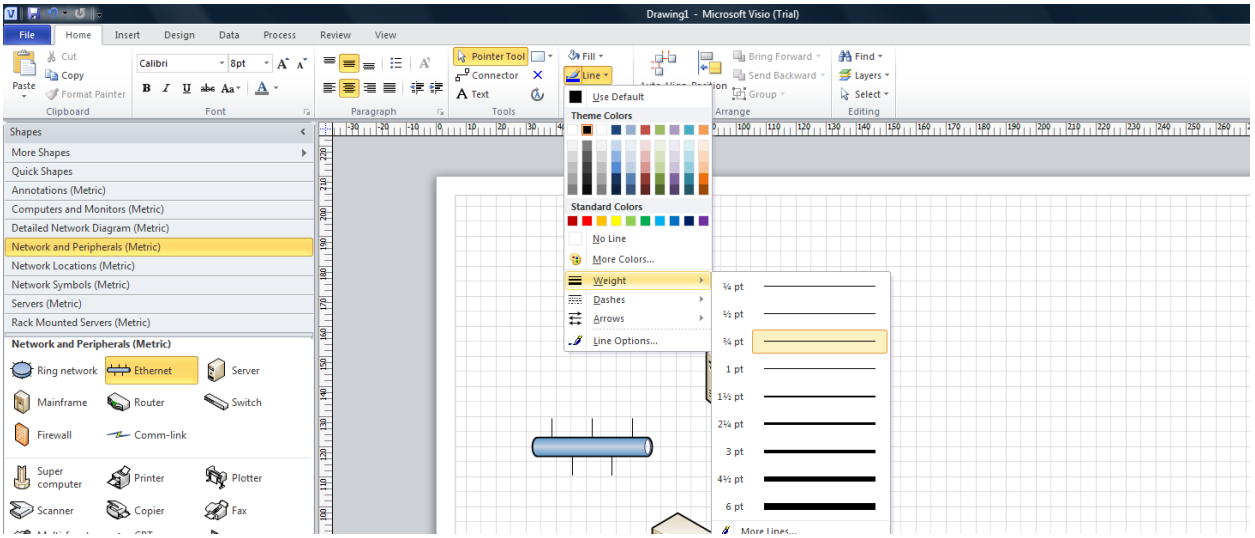
You can select shape using dragging operation:

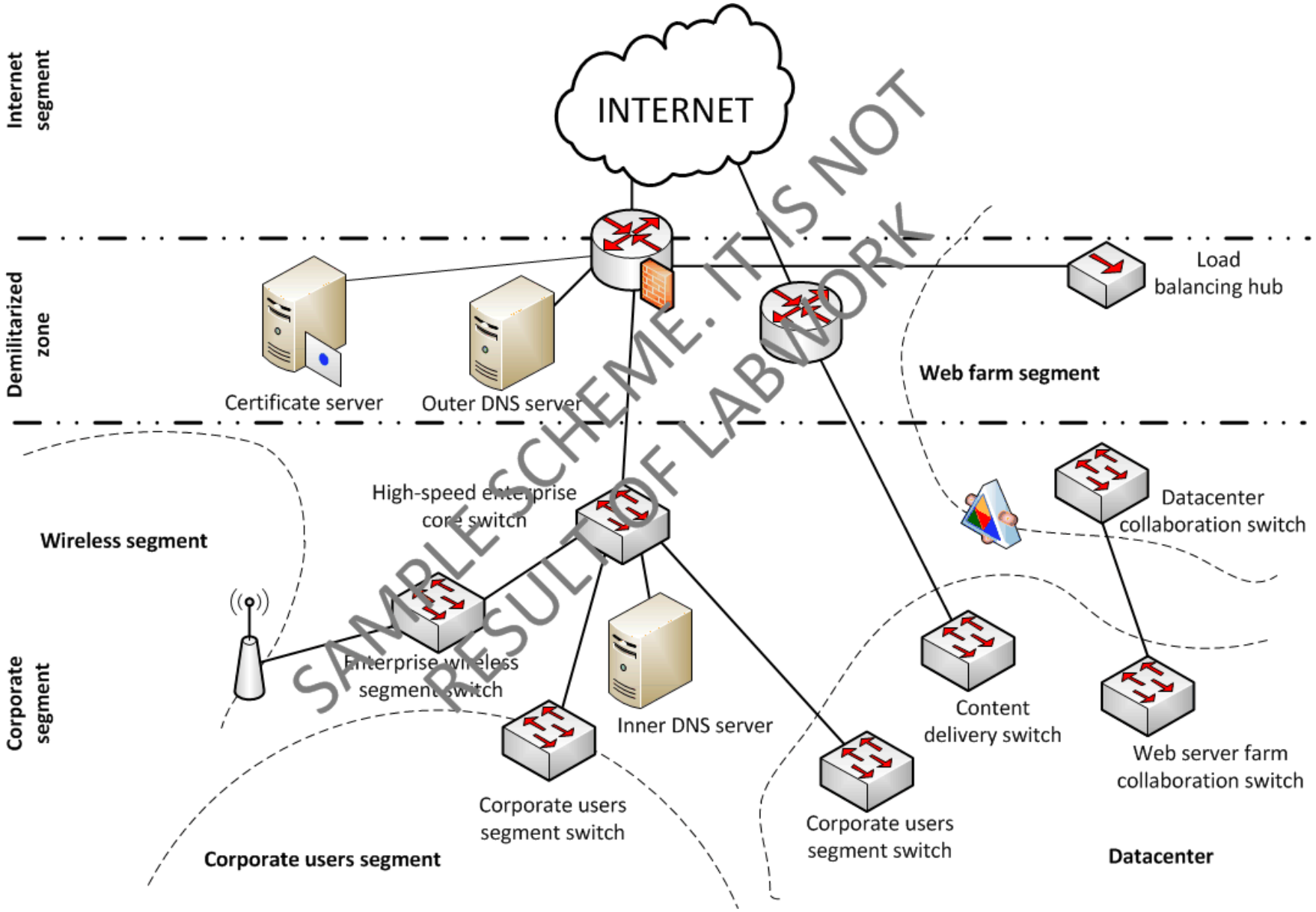


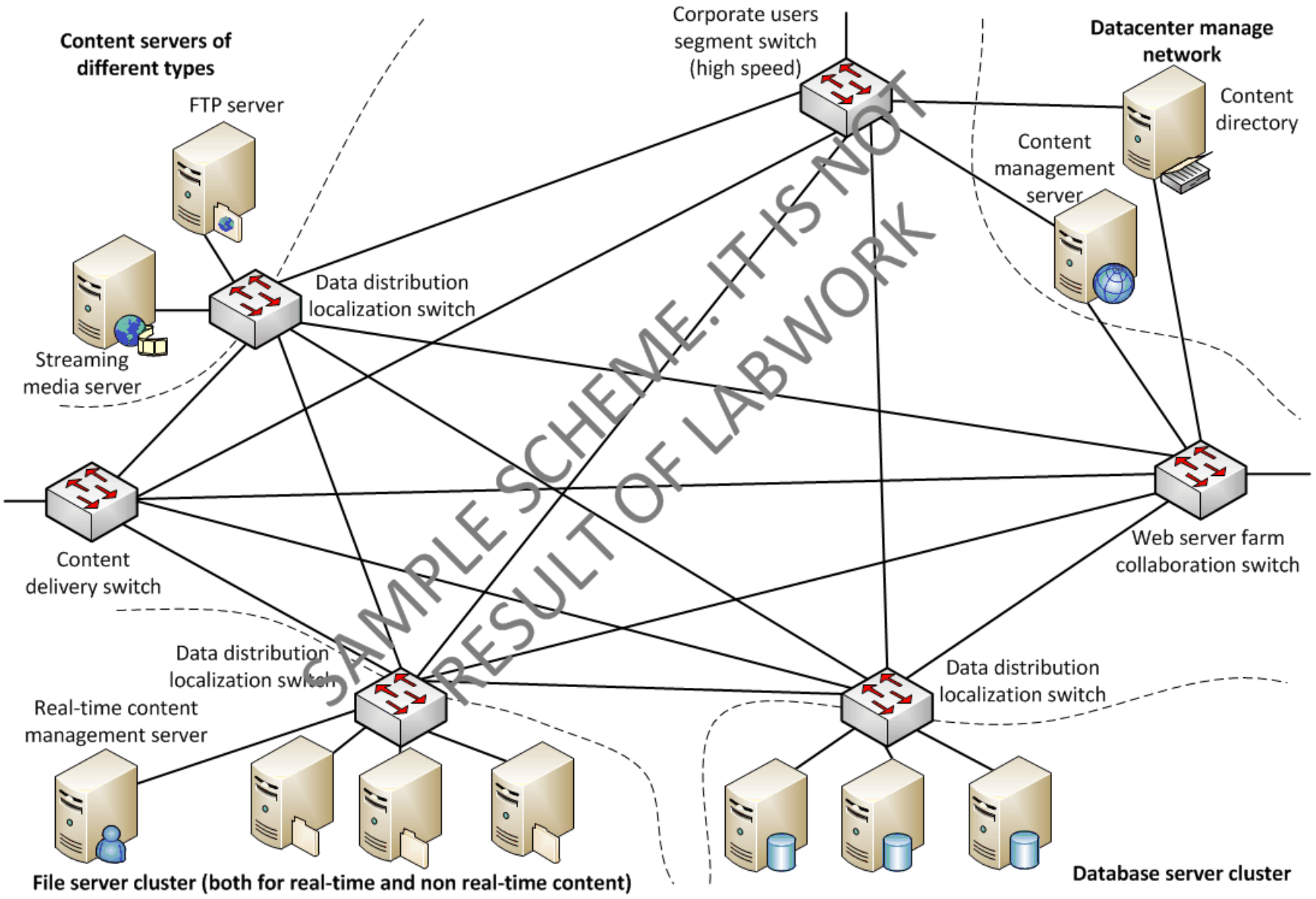
Use lines for network element join:

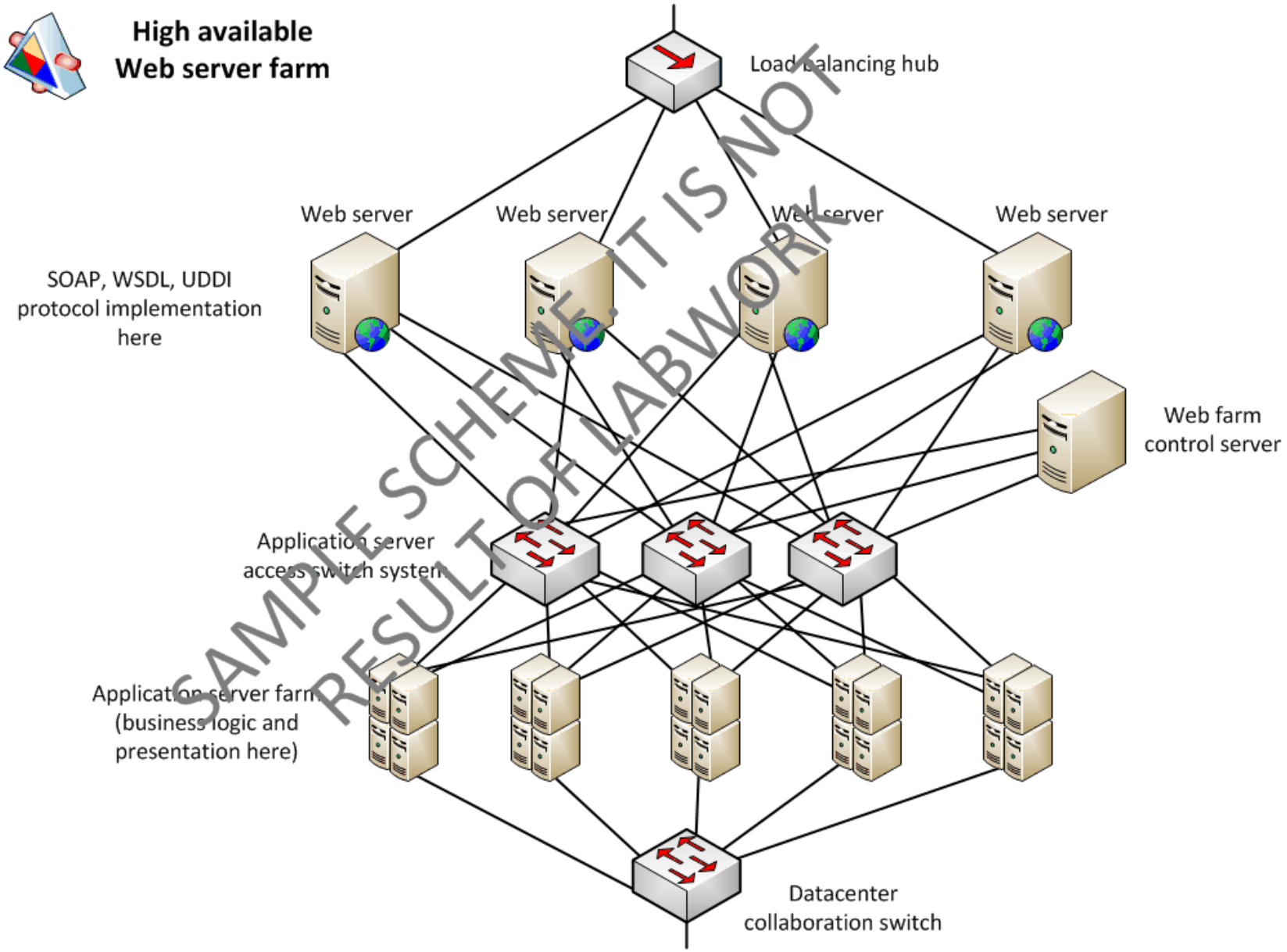


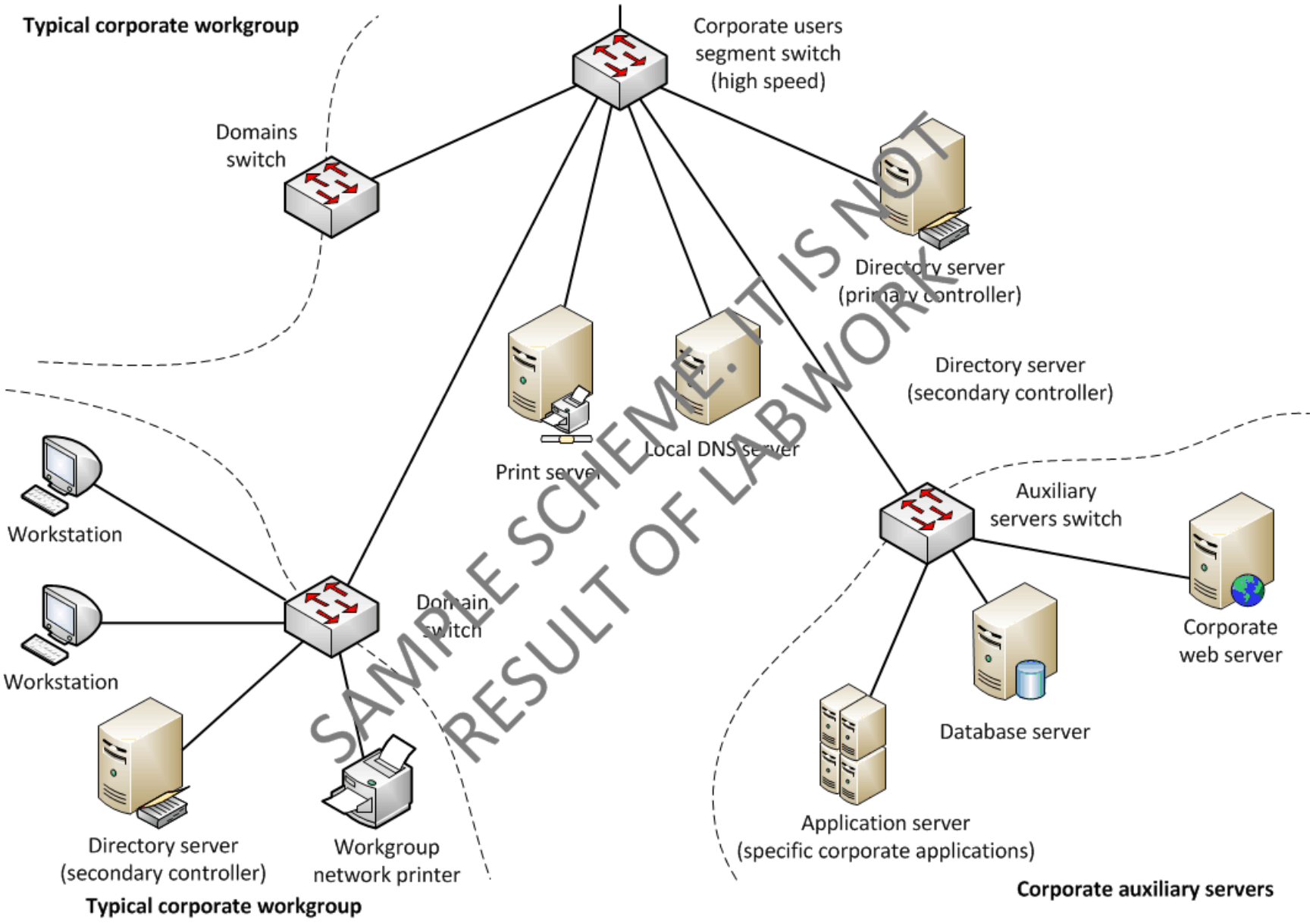
To change line view select line properties menu:




Internet segment
INTERNET
Demilitarized zone
Certificate server
Outer DNS server
Load balancing hub
Web farm segment
Wireless segment
High-speed enterprise core switch
Datacenter collaboration switch
Enterprise wireless segment switch
Inner DNS server
Content delivery switch
Web server farm collaboration switch
Corporate segment
Corporate users segment switch
Corporate users segment switch
Corporate users segment
Datacenter
SAMPLE SCHEME. IT IS NOT RESULT OF LAB WORK


Content servers of different types
FTP server
Streaming media server
Data distribution localization switch
Corporate users segment switch (high speed)
Datacenter manage network
Content directory
Content management server
Content delivery switch
Web server farm collaboration switch
Data distribution localization switch
Real-time content management server
Data distribution localization switch
File server cluster (both for real-time and non real-time content)
Database server cluster
SAMPLE SCHEME. IT IS NOT RESULT OF LABWORK


High available
Web server farm
Load balancing hub
Web server
Web server
Web server
Web server
SOAP, WSDL, UDDI
protocol implementation
here
Web farm
control server
Application server
access switch system
Application server farm
(business logic and
presentation here)
Datacenter
collaboration switch
SAMPLE SCHEME. IT IS NOT
RESULT OF LABWORK


Typical corporate workgroup
Corporate users segment switch (high speed)
Domains switch
Directory server (primary controller)
Directory server (secondary controller)
Print server
Local DNS server
Workstation
Workstation
Domain switch
Auxiliary servers switch
Corporate web server
Database server
Application server (specific corporate applications)
Directory server (secondary controller)
Workgroup network printer
Typical corporate workgroup
Corporate auxiliary servers
SAMPLE SCHEME. IT IS NOT RESULT OF LABWORK