CLOUD COMPUTING

Cloud Application Design

S.Thenmozhi

Department of Computer Applications

CLOUD COMPUTING
Components and Services

- Services are unit of functionality that can run independently

- Services can be deployed separately from the rest of other systems and applications

- Maximum reuse and loosely coupled services

- Used to build resilient systems

- Components are parts of system or application that are designed to work together and deliver a functionality

- Services can be made from components. Components can also use services

Cloud Component Model

- Cloud Component Model is an application design methodology that provides a flexible way of creating cloud applications in a rapid, convenient and platform independent manner

- CCM is an architectural approach for cloud applications that is not tied to any specific programming language or cloud platform

- Cloud applications designed with CCM approach can have innovative hybrid deployments in which different components of an application can be deployed on cloud infrastructure and platforms of different cloud vendors

- Applications designed using CCM have better portability and interoperability

- CCM based applications have better scalability by decoupling application components and providing asynchronous communication mechanisms

CCM Application Design Methodologies

- CCM approach for application design involves:
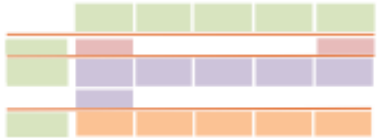 - Component Design
  - Architecture Design
  - Deployment Design

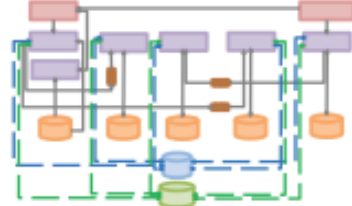# CLOUD COMPUTING
CCM Application Design Methodologies

(a)

(b)
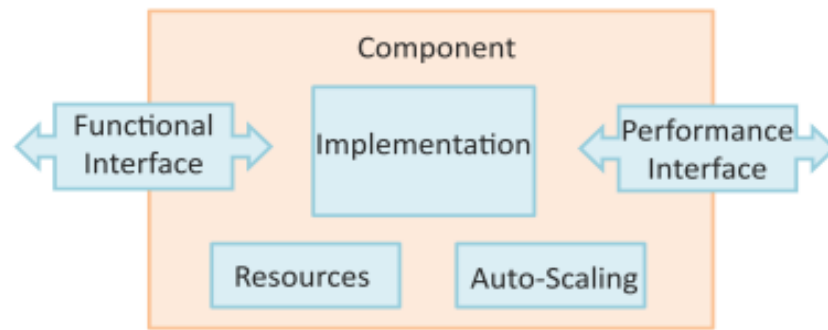
CCM Component Design

- Cloud Component Model is created for the application based on comprehensive analysis of the application's functions and building blocks

- Cloud component model allows identifying the building blocks of a cloud application which are classified based on the functions performed and type of cloud resources required

- Each building block performs a set of actions to produce the desired outputs for other components

- Components report their performance to a performance database through a performance interface
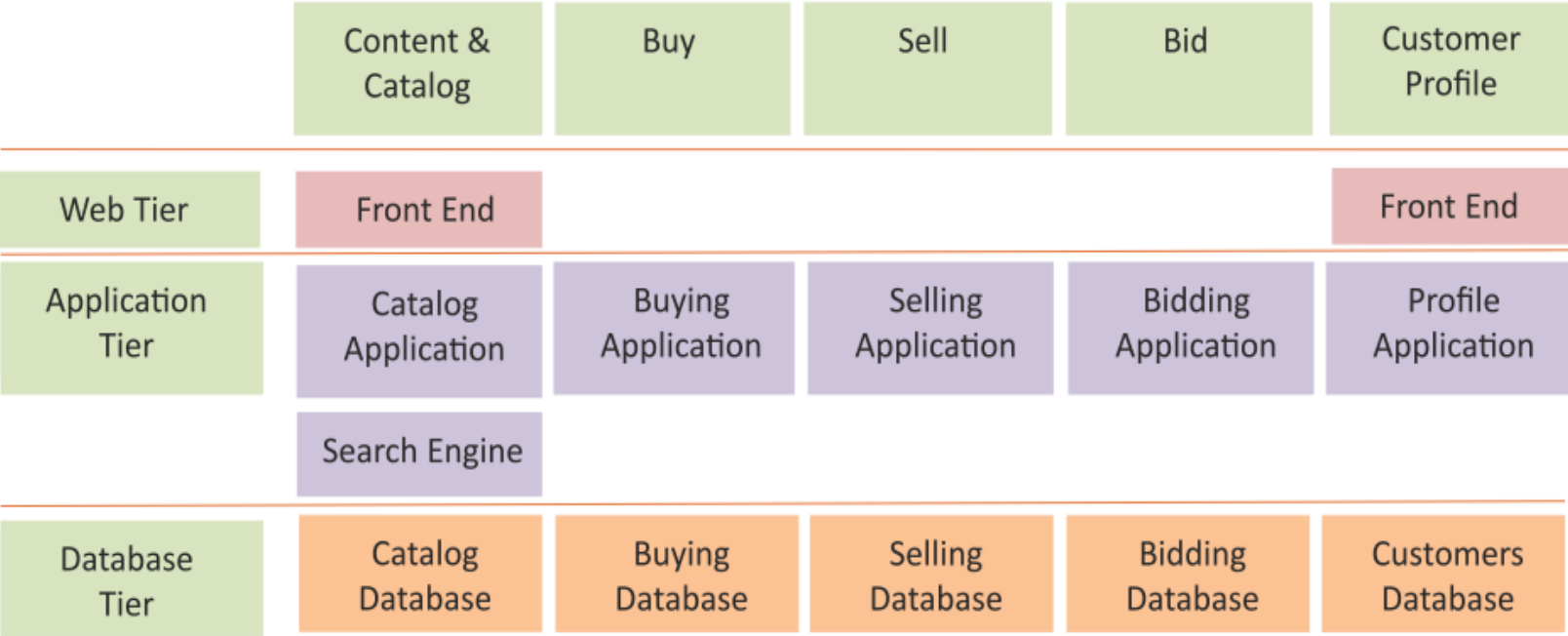
CCM Component Design

- Each component takes specific inputs, performs a pre-defined set of actions and produces the desired outputs

- Components offer their functions as services through a functional interface which can be used by other components

- Components report their performance to a performance database through a performance interface

CLOUD COMPUTING

Component Design for E-Commerce Application

|	Content & Catalog	Buy	Sell	Bid	Customer Profile
Web Tier	Front End				Front End
Application Tier	Catalog Application	Buying Application	Selling Application	Bidding Application	Profile Application
	Search Engine				
Database Tier	Catalog Database	Buying Database	Selling Database	Bidding Database	Customers Database

- In Architecture Design step, interactions between the application components are defined

- CCM components have the following characteristics:

  - Loose Coupling

    - Components in the Cloud Component Model are loosely coupled

  - Asynchronous Communication

    - By allowing asynchronous communication between components, it is possible to add capacity by adding additional servers when the application load increases. Asynchronous communication is made possible by using messaging queues

CCM Architecture Design

- Stateless Design
  - Components in the Cloud Component Model are stateless. By storing session state outside of the component (e.g. in a database), stateless component design enables distribution and horizontal scaling

Architecture Design for E-Commerce Application
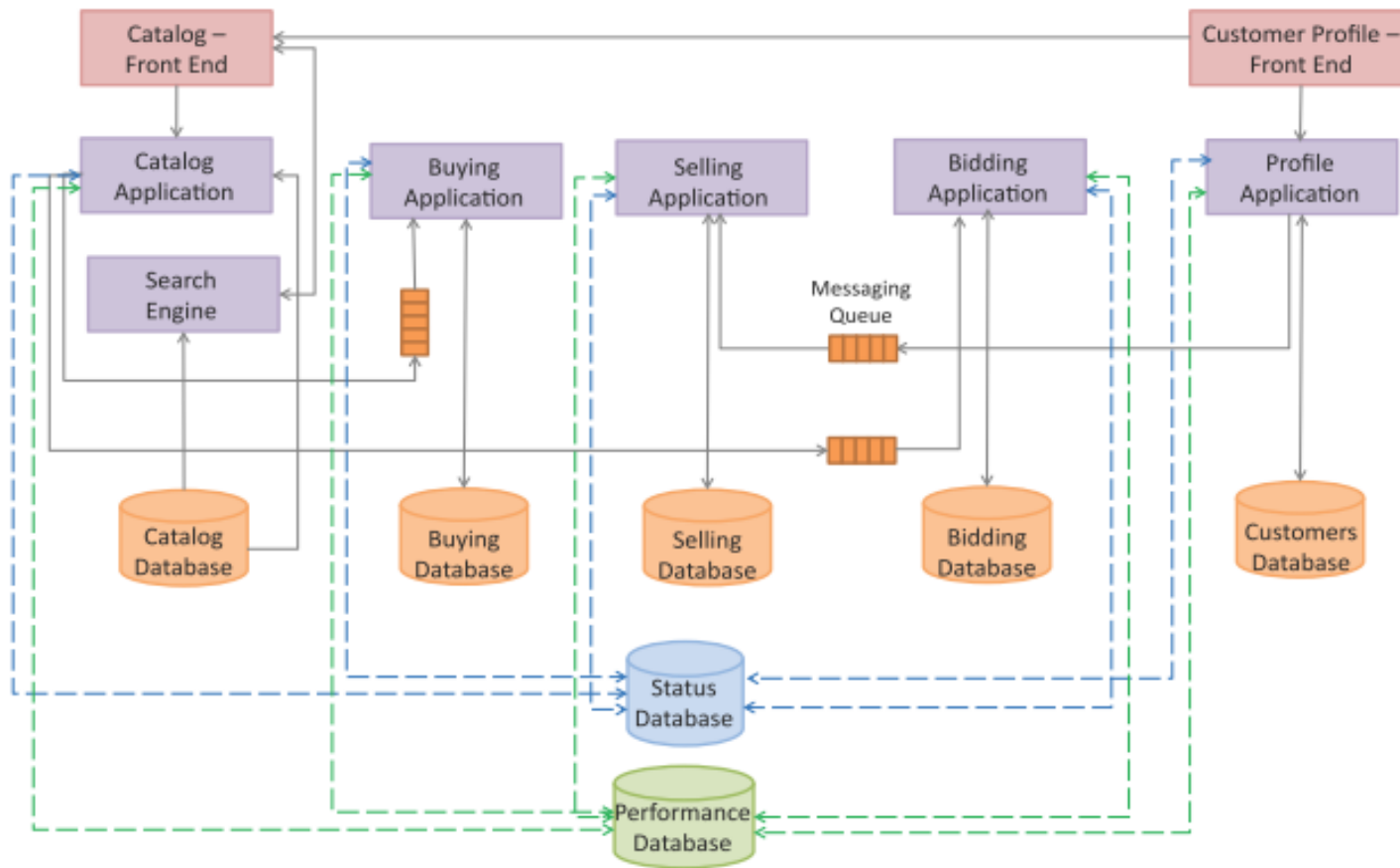
CCM Deployment Design

- In Deployment Design step, application components are mapped to specific cloud resources such as web servers, application servers, database servers, etc

- Since the application components are designed to be loosely coupled and stateless with asynchronous communication, components can be deployed independently of each other

CCM Deployment Design

- This approach makes it easy to migrate application components from one cloud to the other

- With this flexibility in application design and deployment, the application developers can ensure that the applications meet the performance and cost requirements with changing contexts

Deployment Design for E-Commerce Application

SOA Vs CCM
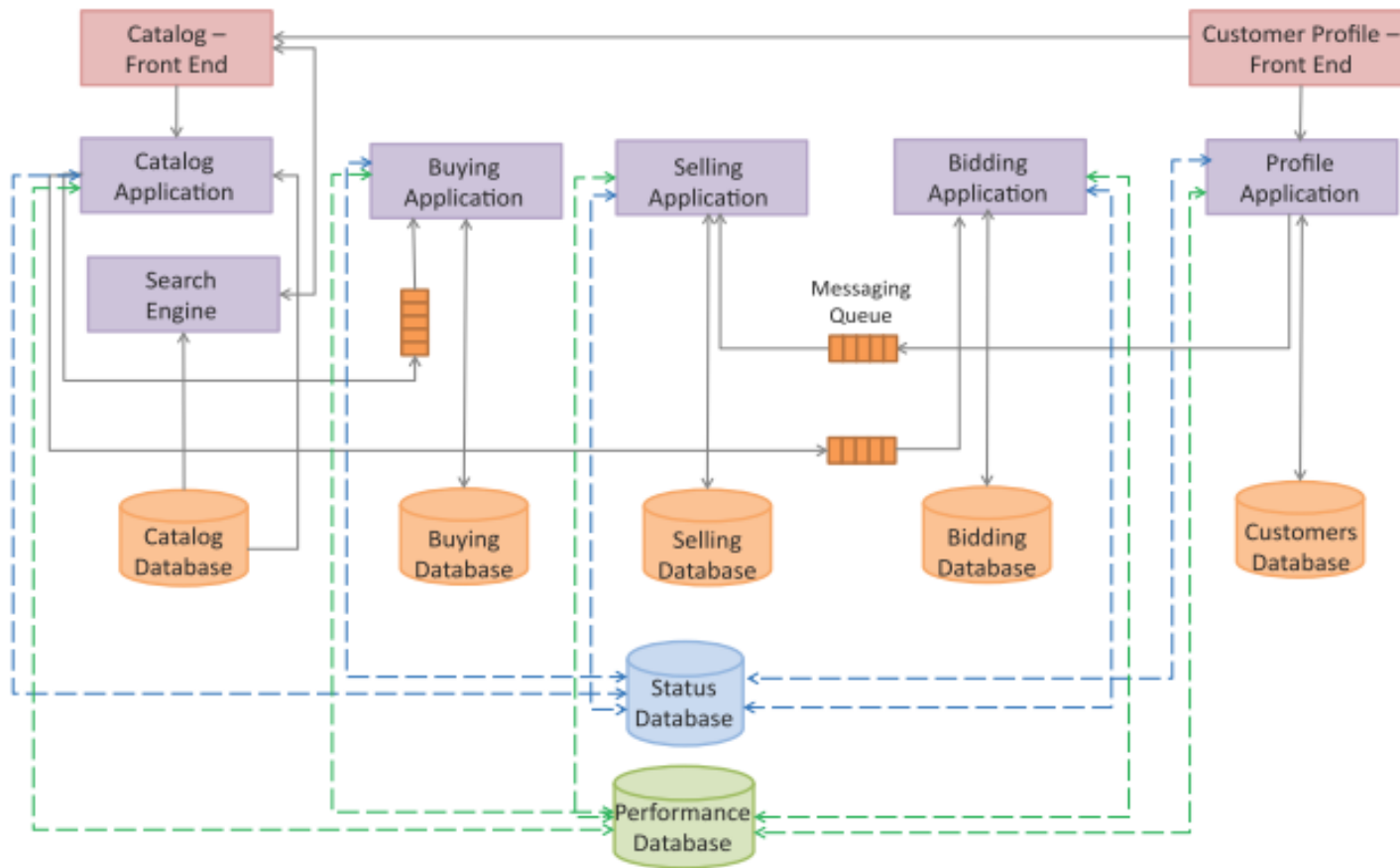
Similarities

	SOA	CCM
Standardization & Re-use	SOA advocates principles of reuse and well defined relationship between service provider and service consumer.	CCM is based on reusable components which can be used by multiple cloud applications.
Loose coupling	SOA is based on loosely coupled services that minimize dependencies.	CCM is based on loosely coupled components that communicate asynchronously
Statelessness	SOA services minimize resource consumption by deferring the management of state information.	CCM components are stateless. State is stored outside of the components.

SOA Vs CCM

Differences

	SOA	CCM
End points	SOA services have small and well-defined set of endpoints through which many types of data can pass.	CCM components have very large number of endpoints. There is an endpoint for each resource in a component, identified by a URI.
Messaging	SOA uses a messaging layer above HTTP by using SOAP which provide prohibitive constraints to developers.	CCM components use HTTP and REST for messaging.
Security	Uses WS-Security , SAML and other standards for security	CCM components use HTTPS for security.

SOA Vs CCM

Differences

	SOA	CCM
Interfacing	SOA uses XML for interfacing.	CCM allows resources in components represent different formats for interfacing (HTML, XML, JSON, etc.).
Consumption	Consuming traditional SOA services in a browser is cumbersome.	CCM components and the underlying component resources are exposed as XML, JSON (and other formats) over HTTP or REST, thus easy to consume in the browser.

THANK YOU

S. Thenmozhi

Department of Computer Applications

thenmozhis@pes.edu

+91 80 6666 3333 Extn 393