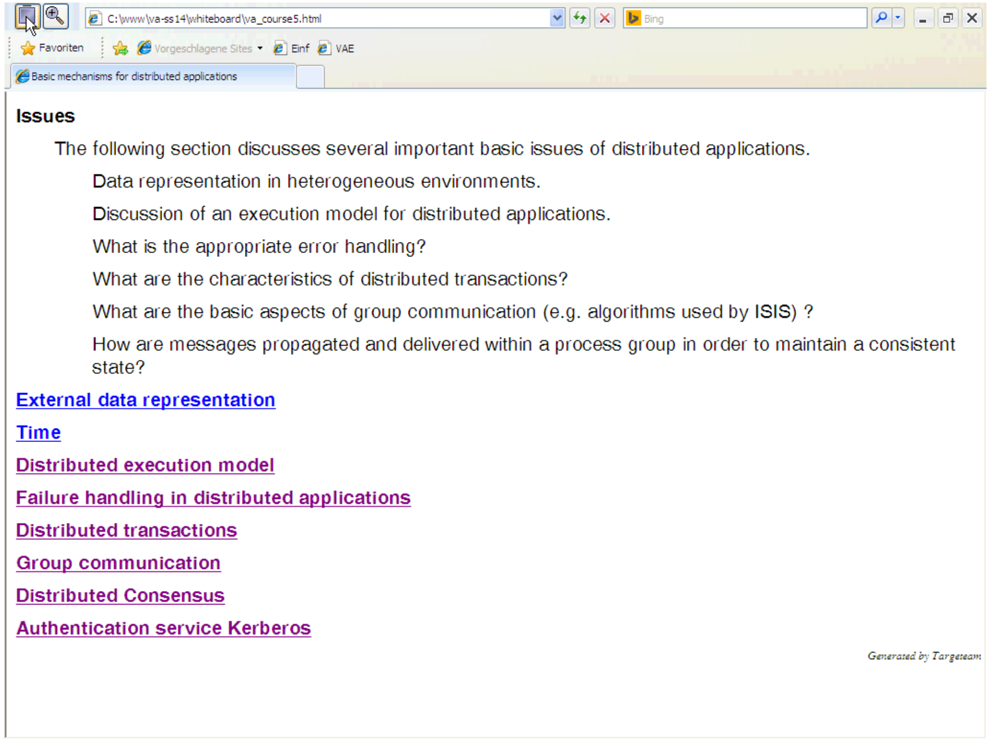
Script **generated by TTT**

Title: Distributed_Applications (16.06.2014)

Date: Mon Jun 16 09:19:20 CEST 2014

Duration: 45:02 min

Pages: 15


C:\www\va-ss14\whiteboard\va_course5.html
Bing
Favoriten
Vorgeschlagene Sites
Einf
VAE
Basic mechanisms for distributed applications

Issues

The following section discusses several important basic issues of distributed applications.

- Data representation in heterogeneous environments.
- Discussion of an execution model for distributed applications.
- What is the appropriate error handling?
- What are the characteristics of distributed transactions?
- What are the basic aspects of group communication (e.g. algorithms used by ISIS) ?
- How are messages propagated and delivered within a process group in order to maintain a consistent state?

External data representation

Time

Distributed execution model

Failure handling in distributed applications

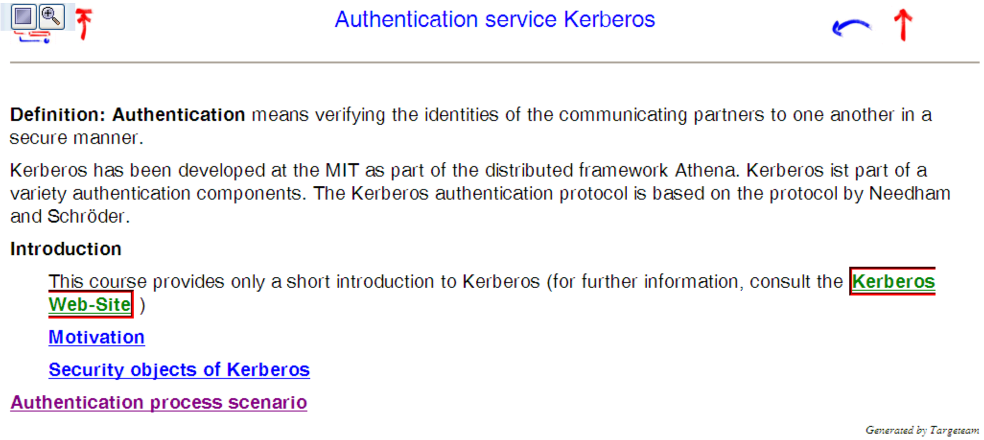
Distributed transactions

Group communication

Distributed Consensus

Authentication service Kerberos

Generated by Targeteam



Authentication service Kerberos

Definition: Authentication means verifying the identities of the communicating partners to one another in a secure manner.

Kerberos has been developed at the MIT as part of the distributed framework Athena. Kerberos ist part of a variety authentication components. The Kerberos authentication protocol is based on the protocol by Needham and Schröder.

Introduction

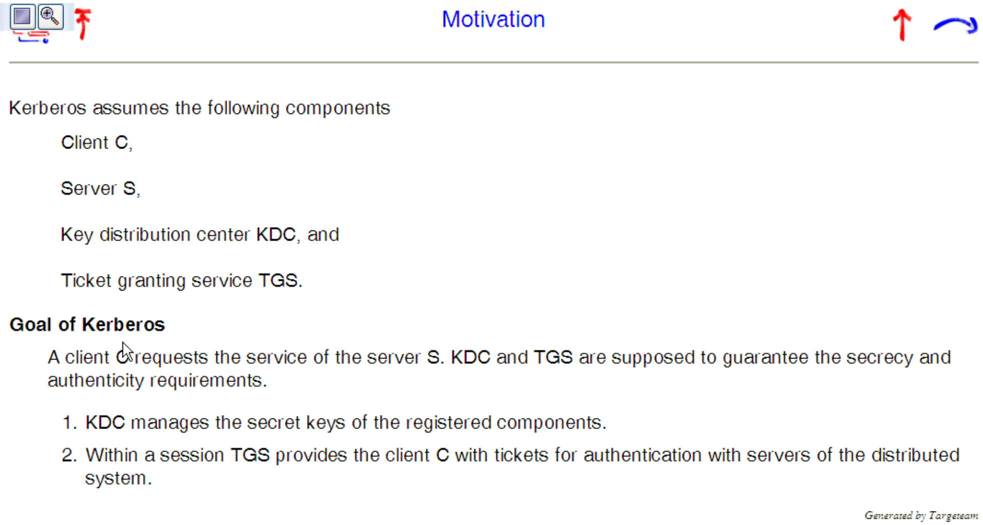
This course provides only a short introduction to Kerberos (for further information, consult the Kerberos Web-Site)

Motivation

Security objects of Kerberos

Authentication process scenario

Generated by Targeteam



Motivation

Kerberos assumes the following components

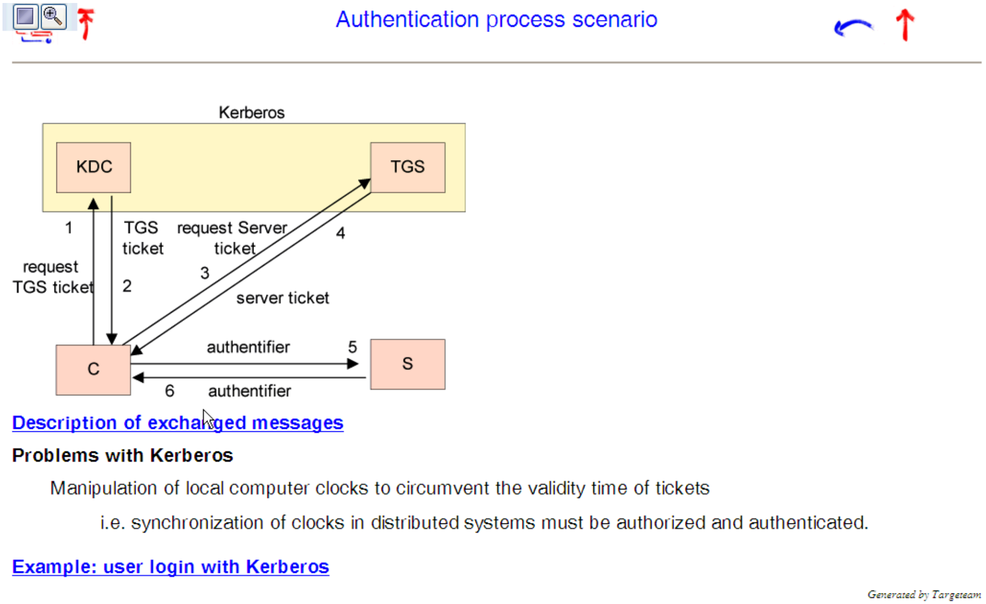
- Client C,
- Server S,
- Key distribution center KDC, and
- Ticket granting service TGS.

Goal of Kerberos

A client C requests the service of the server S. KDC and TGS are supposed to guarantee the secrecy and authenticity requirements.

1. KDC manages the secret keys of the registered components.
2. Within a session TGS provides the client C with tickets for authentication with servers of the distributed system.

Generated by Targeteam



Authentication process scenario

Description of exchanged messages

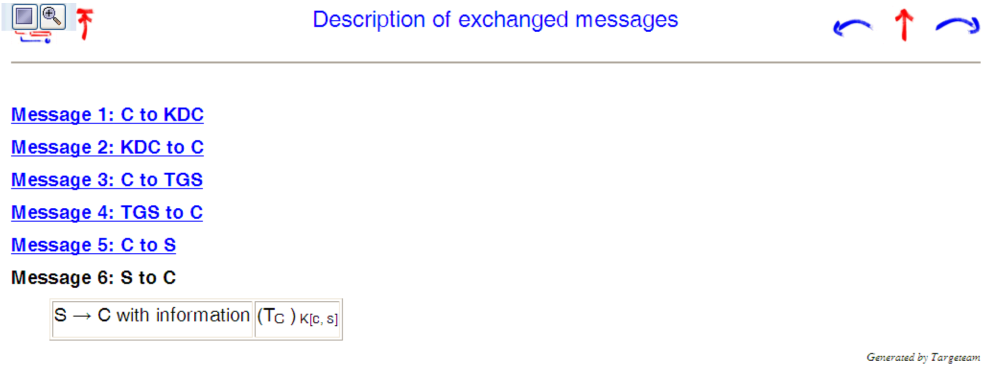
Problems with Kerberos

Manipulation of local computer clocks to circumvent the validity time of tickets

i.e. synchronization of clocks in distributed systems must be authorized and authenticated.

Example: user login with Kerberos

Generated by Targeteam



Description of exchanged messages

Message 1: C to KDC

Message 2: KDC to C

Message 3: C to TGS

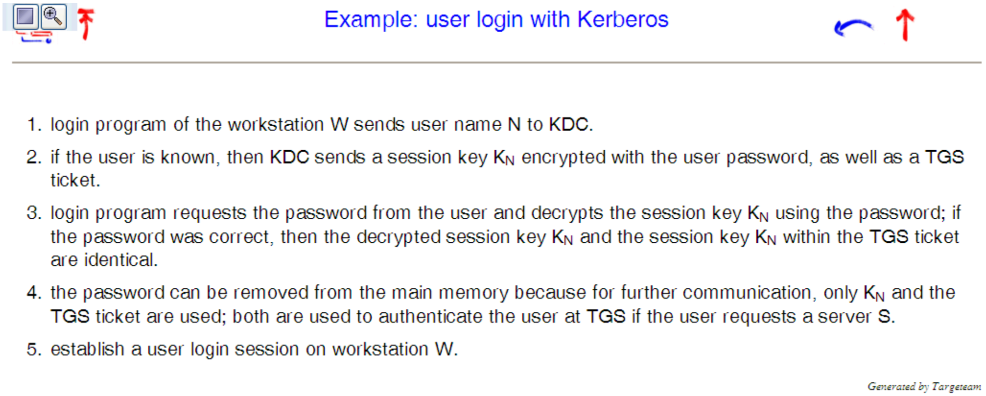
Message 4: TGS to C

Message 5: C to S

Message 6: S to C

S → C with information	$(T_C)_{K[c, s]}$

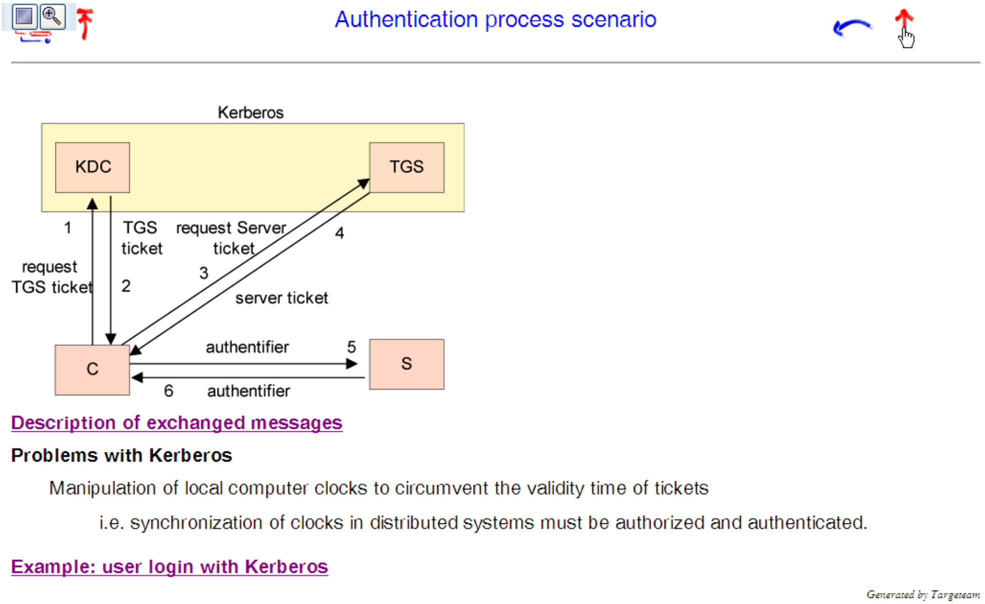
Generated by Targeteam



Example: user login with Kerberos

1. login program of the workstation W sends user name N to KDC.
2. if the user is known, then KDC sends a session key K_N encrypted with the user password, as well as a TGS ticket.
3. login program requests the password from the user and decrypts the session key K_N using the password; if the password was correct, then the decrypted session key K_N and the session key K_N within the TGS ticket are identical.
4. the password can be removed from the main memory because for further communication, only K_N and the TGS ticket are used; both are used to authenticate the user at TGS if the user requests a server S.
5. establish a user login session on workstation W.

Generated by Targeteam



Authentication process scenario

Description of exchanged messages

Problems with Kerberos

Manipulation of local computer clocks to circumvent the validity time of tickets

i.e. synchronization of clocks in distributed systems must be authorized and authenticated.

Example: user login with Kerberos

Generated by Targeteam

Web Services

Web services provide a standard means of communication among distributed software applications based on the Web technology. Standardization by the W3C community.

- Motivation - Example
- Service Oriented Architecture - SOA
- Web Services - Characteristics
- Web Services Architecture
- Simple Object Access Protocol (SOAP)
- Web Services Description Language (WSDL)
- Universal Description, Discovery, and Integration (UDDI)
- REST
- Web Service Composition
- Adopting Web Services
- Mashups

Motivation - Example

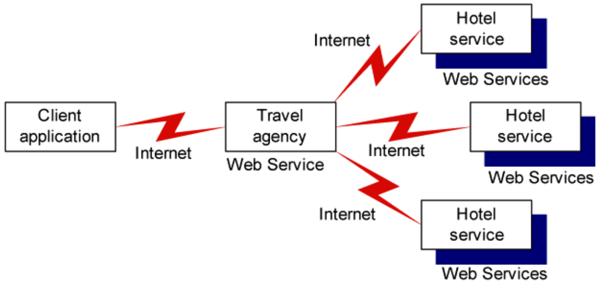
Today, we normally use Web browsers to interact with Web sites

browser names document via URL

request and reply messages encoded in HTML, using HTTP as communication protocol

Web Services generalize this model so that computers can talk to other computers.

Use of Web Services in a distributed travel arrangement application

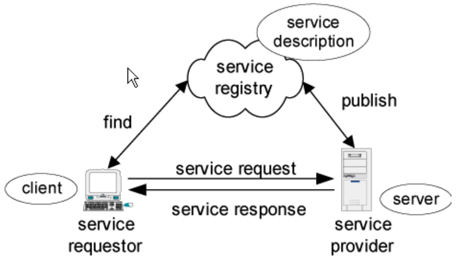


Web Services

Web services provide a standard means of communication among distributed software applications based on the Web technology. Standardization by the W3C community.

- Motivation - Example
- Service Oriented Architecture - SOA
- Web Services - Characteristics
- Web Services Architecture
- Simple Object Access Protocol (SOAP)
- Web Services Description Language (WSDL)
- Universal Description, Discovery, and Integration (UDDI)
- REST
- Web Service Composition
- Adopting Web Services
- Mashups

Characteristics



service is a well defined, self contained function

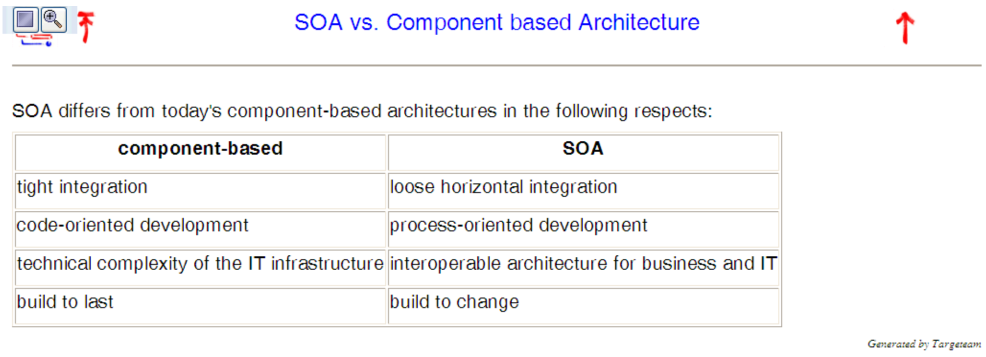
- does not depend on context or state of other services
- manages its own data
- coarse granularity

communication between services

- for data passing and for coordinating activities

focus is on the design of service interface

SOA vs. Component based Architecture

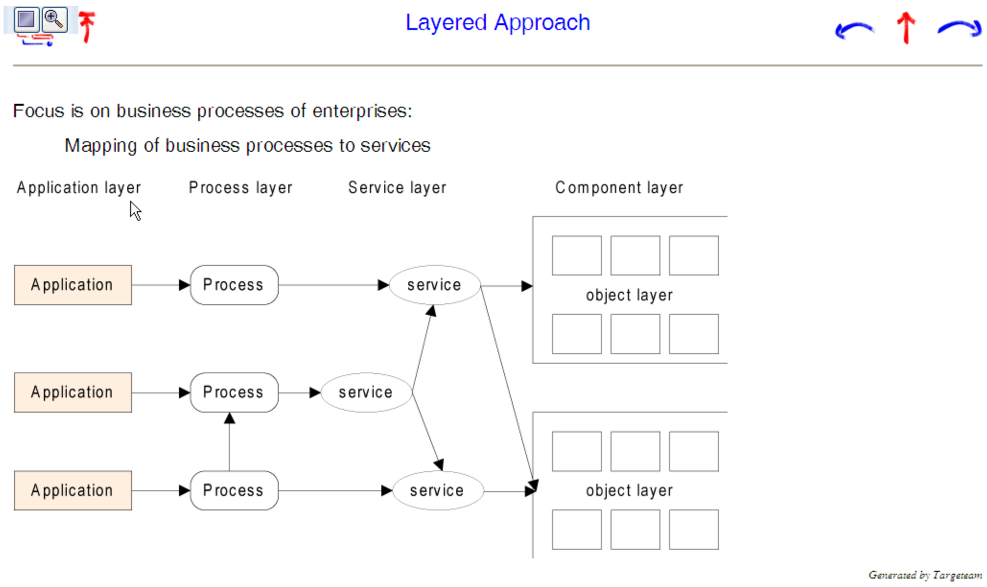


SOA vs. Component based Architecture

SOA differs from today's component-based architectures in the following respects:

component-based	SOA
tight integration	loose horizontal integration
code-oriented development	process-oriented development
technical complexity of the IT infrastructure	interoperable architecture for business and IT
build to last	build to change

Generated by Targeteam

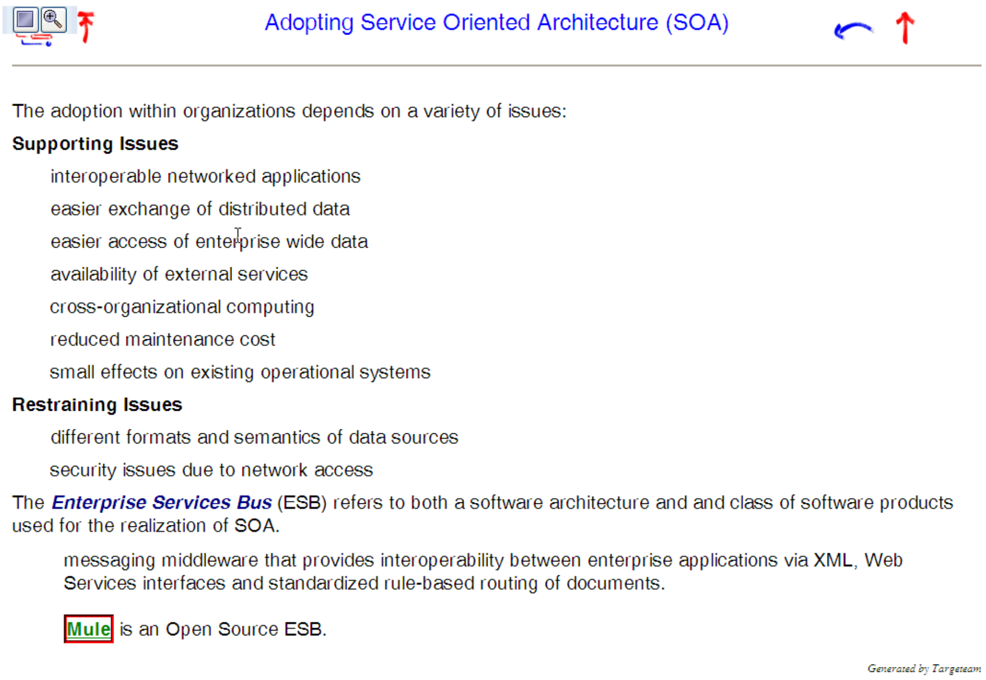


Layered Approach

Focus is on business processes of enterprises:

Mapping of business processes to services

Generated by Targeteam



Adopting Service Oriented Architecture (SOA)

The adoption within organizations depends on a variety of issues:

Supporting Issues

- interoperable networked applications
- easier exchange of distributed data
- easier access of enterprise wide data
- availability of external services
- cross-organizational computing
- reduced maintenance cost
- small effects on existing operational systems

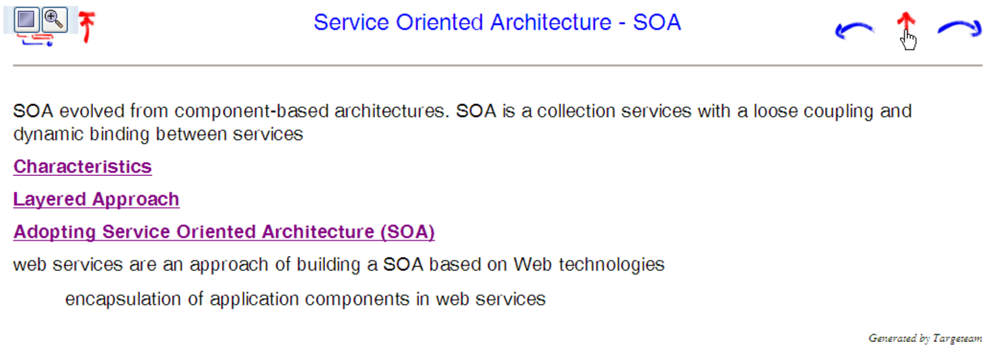
Restraining Issues

- different formats and semantics of data sources
- security issues due to network access

The ***Enterprise Services Bus*** (ESB) refers to both a software architecture and and class of software products used for the realization of SOA.

> messaging middleware that provides interoperability between enterprise applications via XML, Web Services interfaces and standardized rule-based routing of documents.
>
> Mule is an Open Source ESB.

Generated by Targeteam



Service Oriented Architecture - SOA

SOA evolved from component-based architectures. SOA is a collection services with a loose coupling and dynamic binding between services

Characteristics

Layered Approach

Adopting Service Oriented Architecture (SOA)

web services are an approach of building a SOA based on Web technologies

> encapsulation of application components in web services

Generated by Targeteam