Service Oriented Architecture

Application Design – Security

Lecture 9

Dr. Alessio Bonti


Security

Security

CONCEPTS

The challenges of a multi-tenant environment

From the perspective of security, multi-tenant applications pose additional challenges with respect to single-tenant applications.

These challenges are primarily due to keep isolated the different tenants whithin a common computing environment.

Some of the key problems are:

01 PROVIDING A "TENANT-VIEW" ON DATA

02 BALANCING SECURITY WITH PERFORMANCE AND COST

03 FLEXIBLE IDENTITY MANAGEMENT

Security

CONCEPTS

Defining tenant-views on data

A tenant-view data is an abstraction that enables a tenant to see only its own data.

It addresses the following key requirements for multi-tenant applications:

01 TENANT DATA IS ISOLATED FROM OTHER TENANTS DATA

02 THE APPLICATION NEEDS TO BEHAVE AS IF THE TENANTS WERE THE SOLE USER

03 TENANT DATA MUST BE PRIVATE AND PROTECTED FROM UNWANTED ACCESS

04 TENANTS WANT TO HAVE COMPLETE CONTROL ON THEIR DATA

Security

CONCEPTS

Defining tenant-views on data

There exist different options to implement such views each of them with different trade-offs. The most common are the following:

01

BRING YOUR OWN SUBSCRIPTION

This solution implies that the tenant provide the multi-tenant application provider with subscriptions and accounts for storage. This solution provides the highest degree of isolation for the data, since the information of each tenant is maintained under a different subscription.

PROS

- *the tenant view is simply implemented by keeping information about the subscriptions to use*
- *the resulting cost of the application is smaller because storage cost are paid by the tenant*
- *tenant can reuse their existing subscriptions*
- *tenant have more easily access to their own data and full control on it*

CONS

- *it is not a "turn-key" solution, but it has dependencies on tenant owned resources*
- *the tenant must bear additional costs and trust the provider usage of their subscriptions*

Security

CONCEPTS

Defining tenant-views on data

(… continued)

02

SEPARATE STORAGE PER TENANT

This scenario includes using either a separate subscription or database/storage container per tenant but the accounts are owned by the provider of the multi-tenant applications. This solution still provides a good level of isolation between tenants.

PROS

- *the tenant view is constituted by either subscription or storage target information*
- *the level of isolation is as good as the previous solution*
- *the application comes with all the required resources ("turn-key" solution)*

CONS

- *the resulting cost of the application may be higher as it needs to absorb storage costs*
- *the access to tenant data is filtered by the application*
- *the solution might be too expensive for the provider of the application*

Security

CONCEPTS | Defining tenant-views on data

(… continued)

03

SHARED STORAGE AND LOGICAL VIEWS

This scenarios implies the user of a common subscription or a common set of subscriptions that is used to serve multiple tenants. More specifically, different tenant share the same database schema or the same storage container.

PROS

- *the application is a "turn-key" solution*
- *the application provider can optimise at best costs on storage*

CONS

- *the implementation of the tenant view can be more complex and prone to security issues*
- *the security must be enforced by the logic of the application*
- *operations such as tenant backups and replication operate on a portion of the data*
- *if there are different types of SLA on data they might become harder to implement*

Security

CONCEPTS

Applying the right level of protection

Within an application there might be different pieces of information that require different level of protection, it is up to the application designer to find the right balance.

01

DATA THAT MUST BE PROTECTED

For information that must be protected (i.e. confidential data, credit card details, personal sensitive information and personal health information) consider:

ENCRYPTION

This approach includes several techniques, the most common we can think is encrypting tenant data and connections. To improve security and isolation we can use a different key for a different tenant.

ADVANCED TECHNIQUES

These aim at reducing the risk when the storage is compromised. Consider splitting credit card details into different storage accounts to reduce the chance of accessing the entire information.

Security

CONCEPTS | Applying the right level of protection

(… continue)

02

DATA THAT WE WOULD LIKE TO GIVE RESTRICT ACCESS TO

There are cases when we would like to restrict the access to information, but unauthorised accesses might not create a huge damage. This case applies to non-sensitive information which we would still keep secret but we're not willing to secure as discussed previously.

In this case a sufficient level of protection might be achieved by disclosing the access details only to the people we want to access that information.

SHARED ACCESS SIGNATURES (VALET KEY PATTERN)

A shared access signature is a hard to guess URL that provides access to a resource to anyone knowing the URL. The access to the resource can be temporary.

Security

FOCUS POINT

A few considerations

These are just some of the many that aspects apply to securing applications that meet the expectations of users in a multi-tenant environment.

There is one general observation that we always need to keep in mind:

SECURITY AS A TRADEOFF

The implementation of security measures is always a tradeoff between how valuable we consider the information we want to protect and how valuable our attackers consider it.

The value we attribute to the information we want to protect influences what we're willing to do protect it (i.e. the sophistication and cost of our security implementation).

The value our attackers attribute to it defines the extents they're willing to go to break our security implementation.

Security

CONCEPTS

Flexible identity management

Identity management is an essential component of multi-tenant applications. It is at the basis of multi-tenant application consistency. If we are unable to identify the tenant each request belongs to, we are unable to ensure isolation in a multi-tenant environment.

There are two major approaches for identity management:

01

"IN HOUSE" IDENTITY MANAGEMENT

The users and the verification of their identities is completely managed within the multi-tenant application.

02

FEDERATED IDENTITY MANAGEMENT

The verification of user credentials is left to a third party and the multi-tenant application retains only the information needed for the application (e.g. is valid user? corresponding role, ..).

Security

CONCEPTS

In-house identity management

In-house identity management refers to an implementation of any framework for identity management that is completely carried out within the application space (or eventually uses platform services).

We're not interest in the different types of implementation, but more in what type of advantages / disadvantages it gives:

PROS

- *the application is a "turn-key" solution*
- *it can be fully customised to the needs of the application*
- *it can be advantageous for organisations that cannot expose identity management*

CONS

- *it is additional work for a multi-tenant application developers*
- *the tenants might need to remember additional credentials specific to the application*

Security

CONCEPTS

Federated identity management

Federated identity management relies on a third party for verifying the identity of the user and its associated roles. The application uses the information returned by the "identity provider" (i.e. identity claims) to create an identity within the applications.

Provider examples: Google, Facebook, your organisation.

PROS

- *users can user a familiar identity mechanism, without learning new credentials*
- *facilitates the integration with an existing user base, without recreating it*
- *it (lightly) reduces the size of application codebase and its maintenance*

CONS

- *integration might become hard if the identity provider does not implement a standard*
- *the application needs to implement role mapping*
- *the solution might not be beneficial if the identity provider does not support roles and the application requires role management.*

Security

CONCEPTS

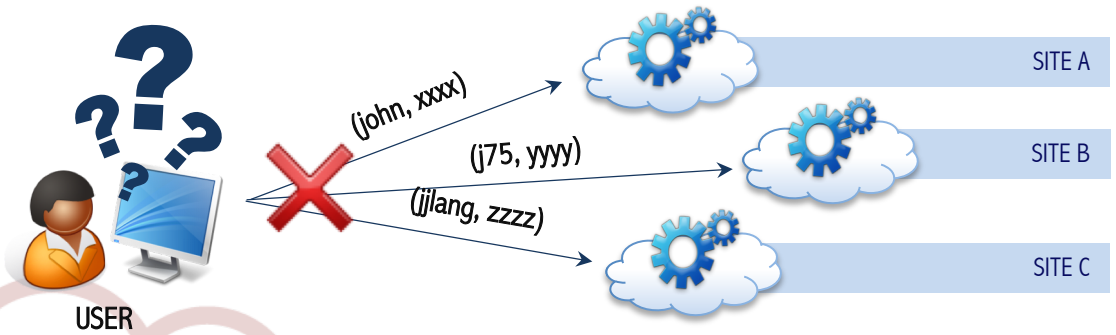
Federated identity management

How does it work?

There exist multiple techniques to implement federated identity management, we will focus on those that are based on authentication.

INTRODUCING SINGLE SIGN ON (SSO)

Federated Identity Management is a set of practices that provides means to share someone's digital identity across organisations. When the specific techniques focus on authentication mechanisms a popular solution is based on Single Sign On (SSO).

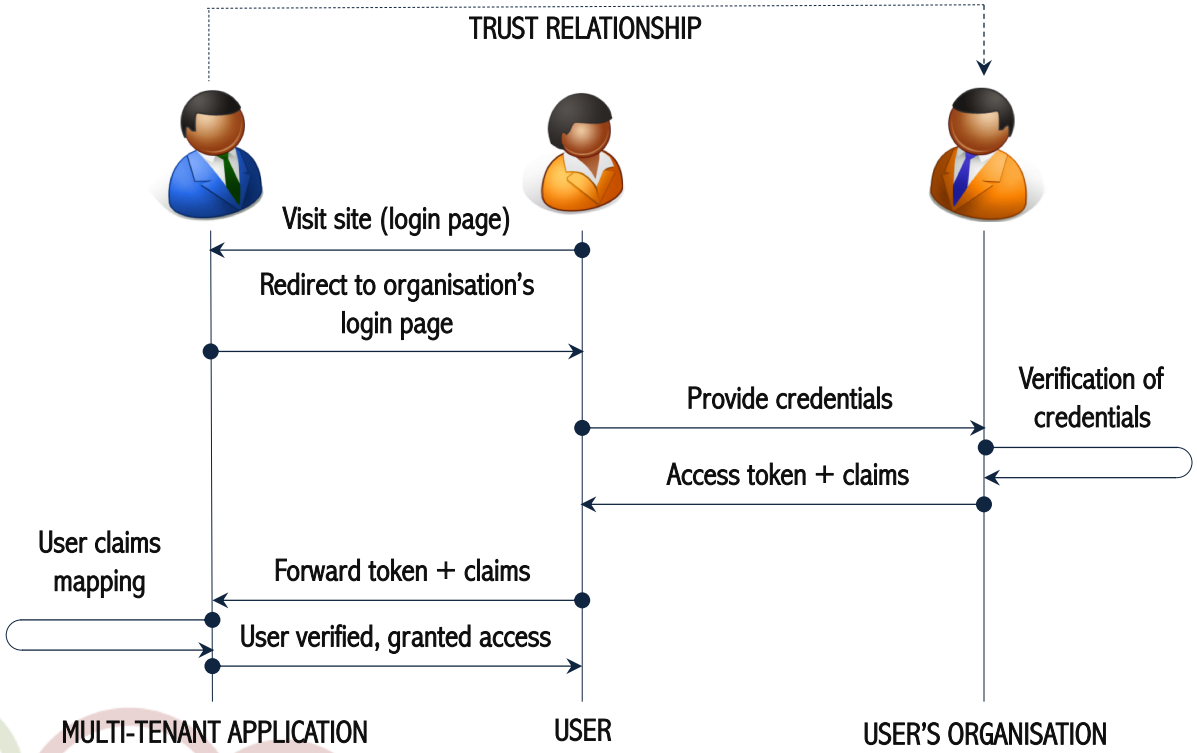


Security

CONCEPTS

Federated identity management

Example with an identity provider.

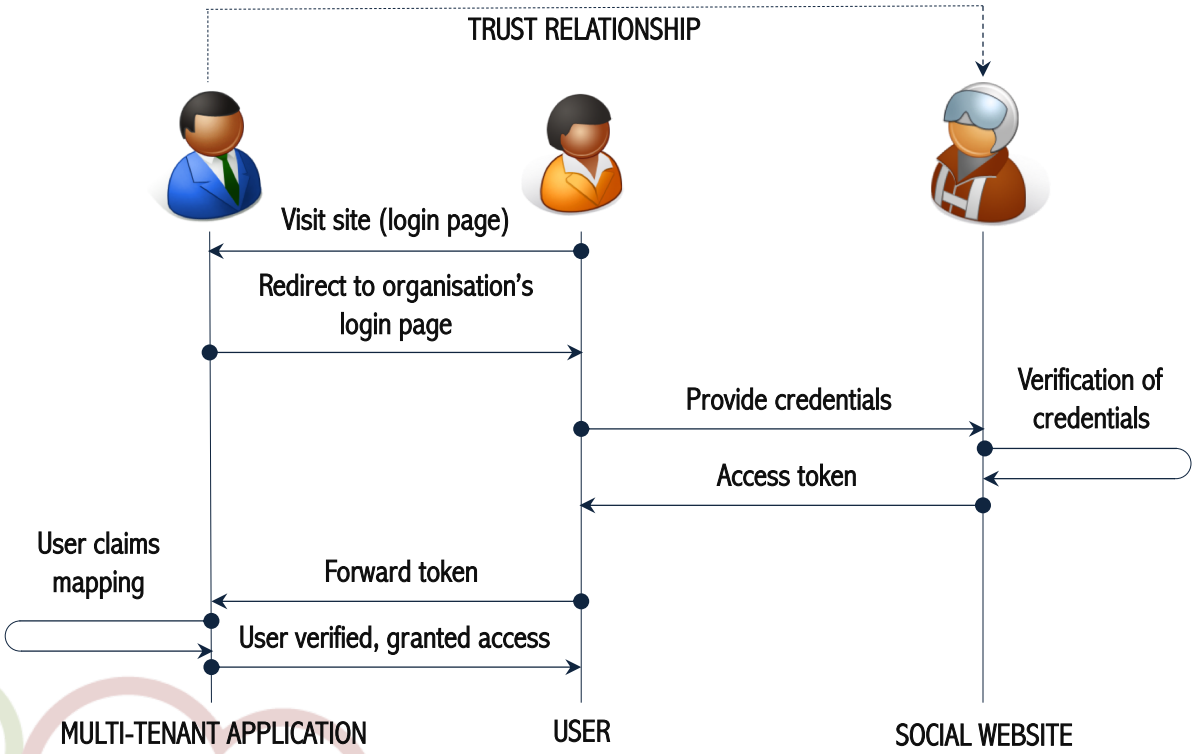


Security

CONCEPTS

Federated identity management

Example with a "social identity" provider. The process is similar, but the authentication token does not contain additional claims.



Security

CONCEPTS

Federated identity management

What types of technologies and standards are used to implement federated identity and SSO solutions?

01 LDAP (LIGHTWEIGHT DIRECTORY ACCES PROTOCOL) *https://tools.ietf.org/html/rfc4511*

02 OAUTH *http://oauth.net/2/*

03 OpenID CONNECT *http://openid.net/connect/*

04 WS-FEDERATEDION *http://docs.oasis-open.org/wsfed/federation/v1.2/ws-federation.html*

05 SAML (SECURITY ASSERTION MARKUP LANGUAGE) *https://wiki.oasis-open.org/security/FrontPage*

Security

APPLICATIONS

A look to our application…

How do we apply these concepts to the design of the security of the "Your Events" application?

01 LOGICAL VIEWS

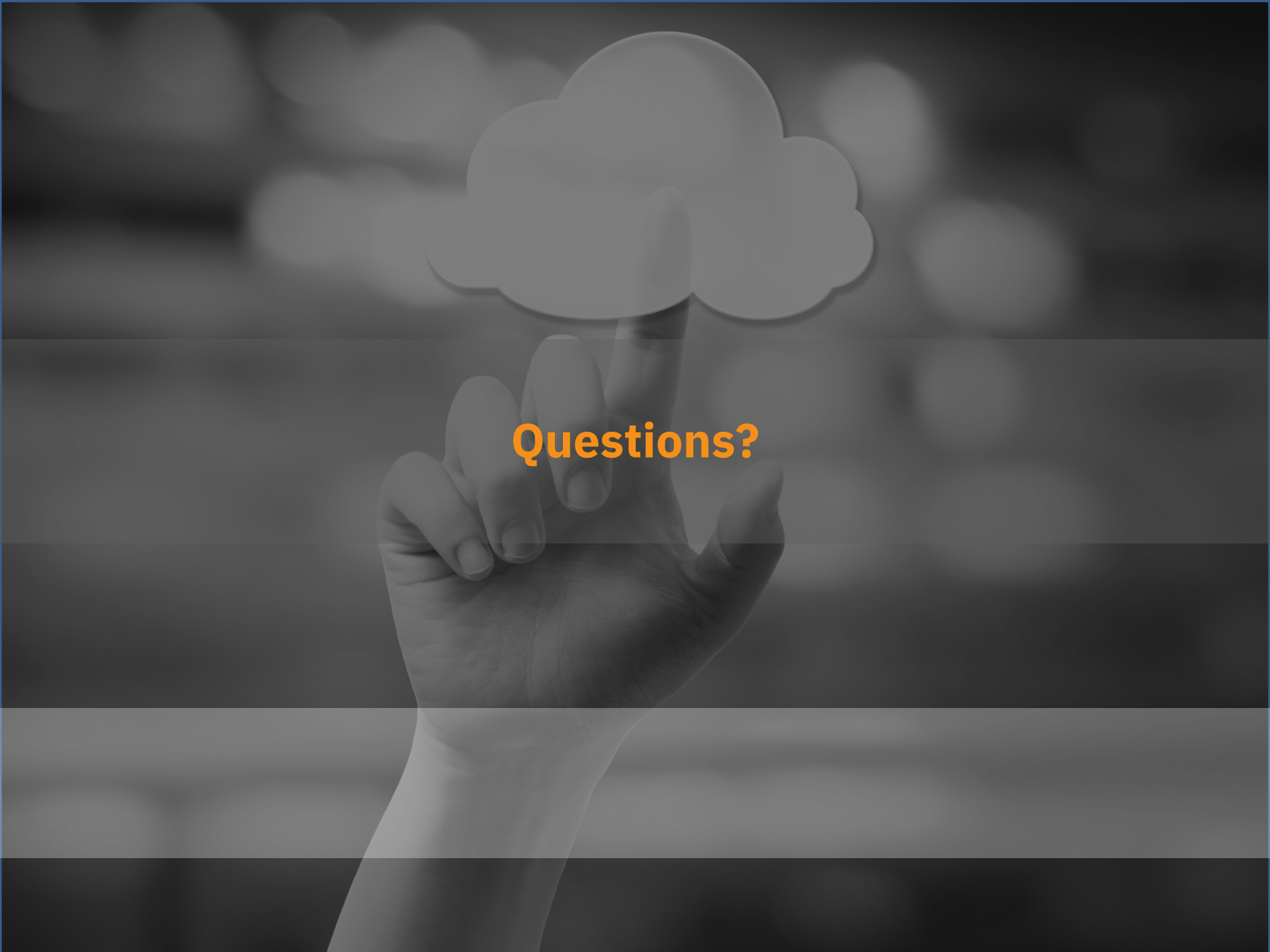
Cloud Dynamics will implement logical tenant views over a shared storage to maximise the use of their own storage subscription. Different tenant groups may be handled with different accounts.

02 ENCRYPTION AND SHARED ACCESS SIGNATURES

Cloud Dynamics will encrypt the information contained in the storage. Premium subscriptions will be encrypted with a separate key for each tenant. The application will use shared access signatures for images and thumbnails.

03 IDENTITY MANAGEMENT

Premium subscribers can leverage their own identity provider, while standard subscribers will be using a built-in user identity management infrastructure with the application.


Questions?