Securing Async Service Communication

Kevin Dockx
ARCHITECT

@KevinDockx https://www.kevindockx.com

Coming Up

Approaches to service bus security

Securing asynchronous communication

Choosing the right security approach for your use case

Approaches to Service Bus Security

Transport should be secured (TLS or other means)

Authentication with the bus

Granular (message) authorization

Granular (Message) Authorization

“Service 1 will only accept messages from service 2 or service 3”

“Service 1 will only accept messages from user A”

Azure Service Bus

Shared Access Signature (SAS)

Azure AD

- Access token (dis)allows access to the bus
- Depending on the token, access to the specified resource is authorized
- Levels: subscription, resource group, service bus namespace

Approaches to Service Bus Security

Authentication approaches are often specific to a bus implementation

- Azure Service Bus, NServiceBus, … approach this differently

Granular Message Authorization

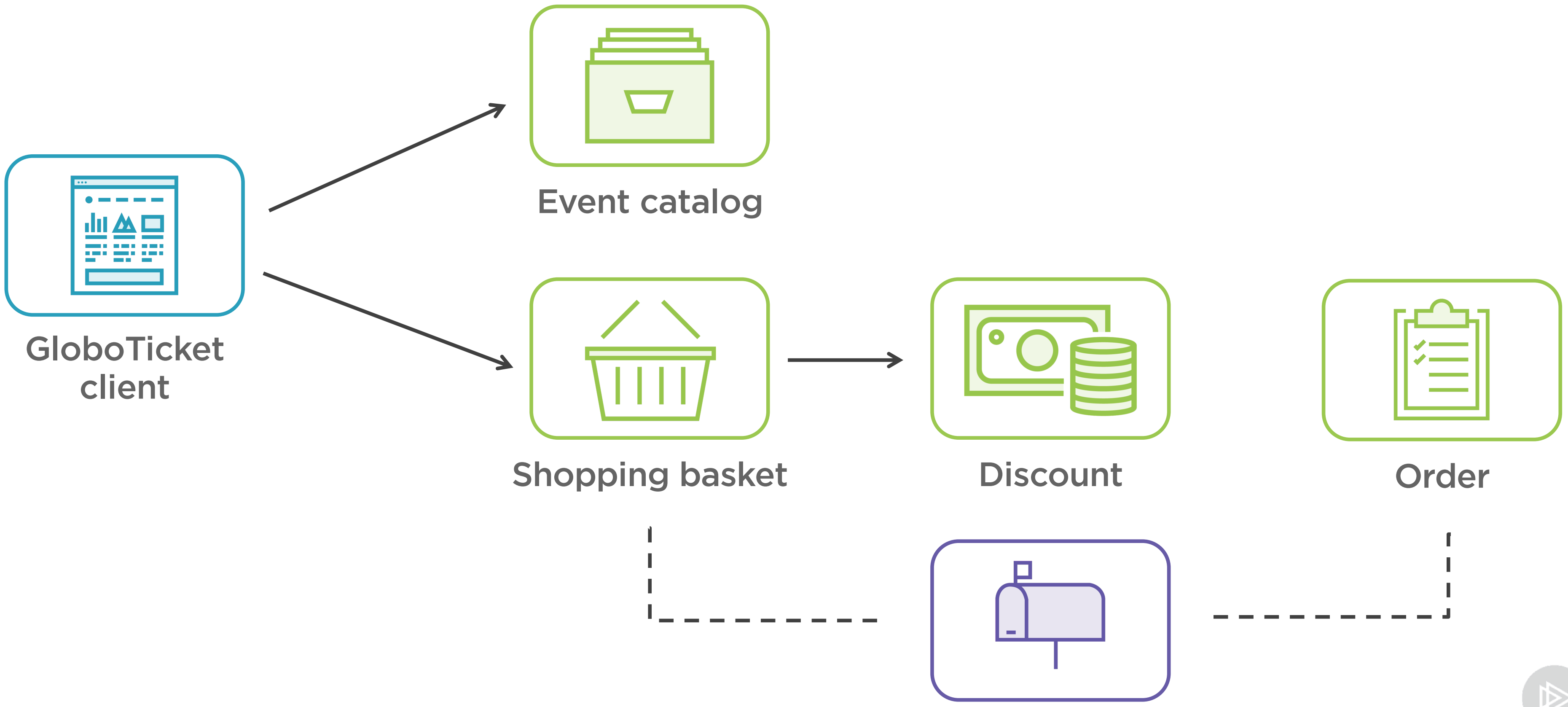
Send a token together with each message

When the message is received, extract and validate the token

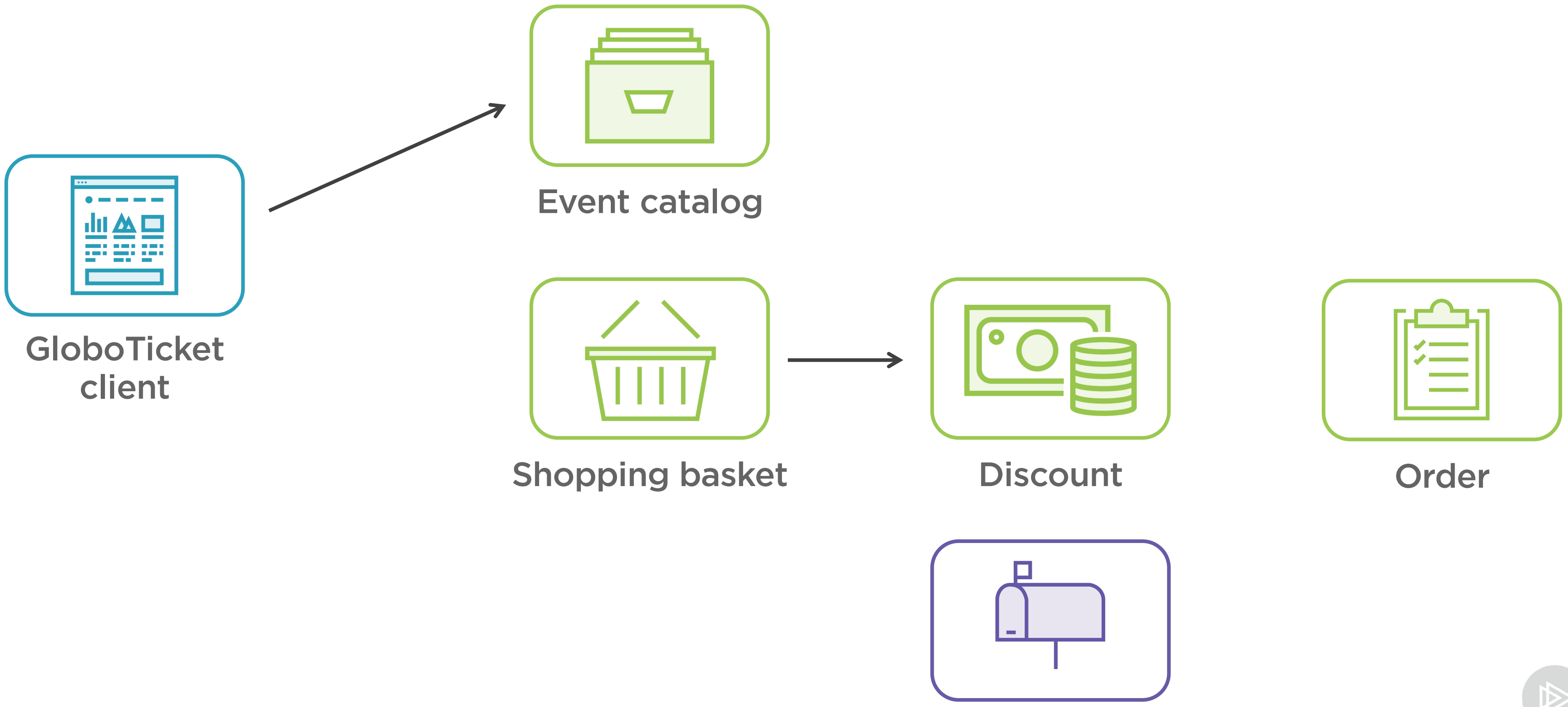
- (Dis)allow access depending on the result of this validation

Not tied to a specific service bus implementation

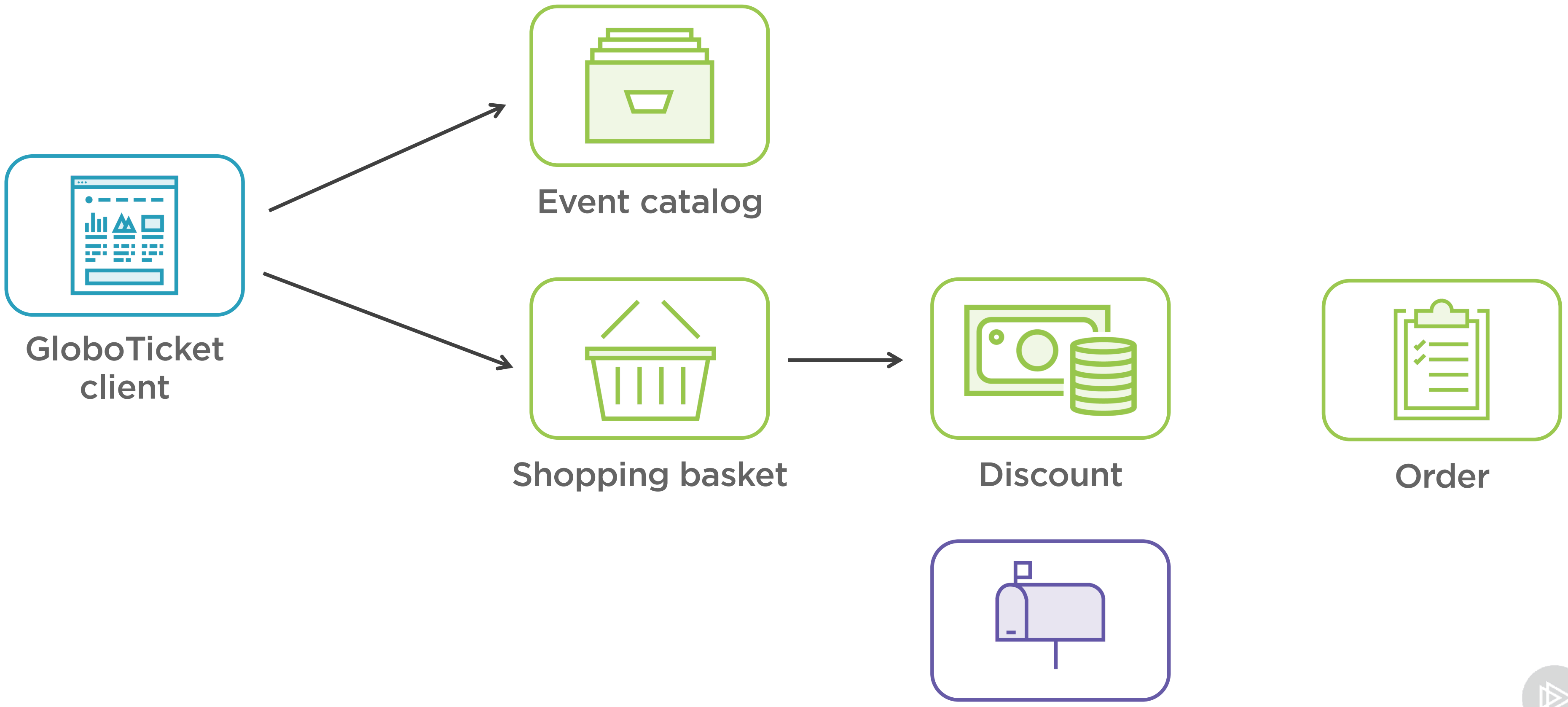
Introducing the Upcoming Demos



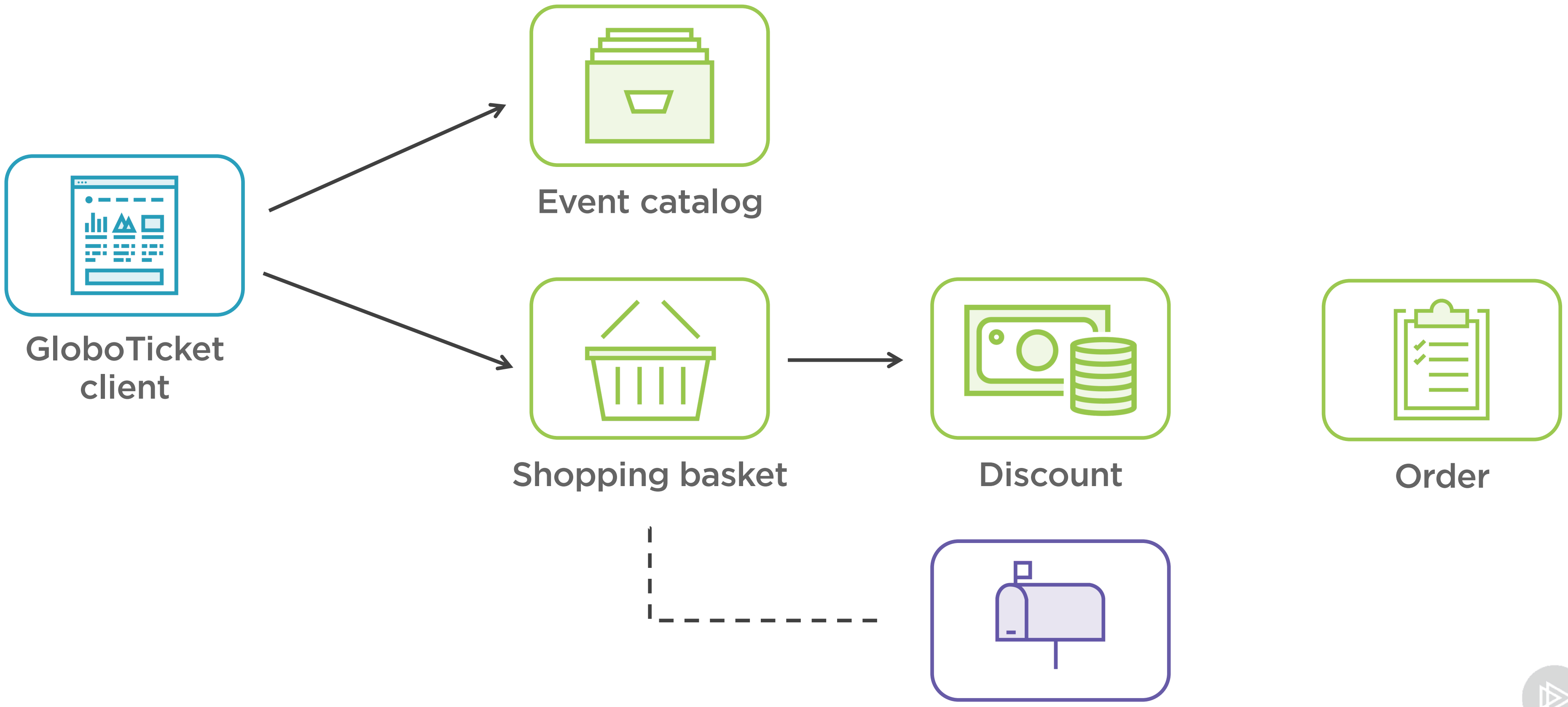
Introducing the Upcoming Demos



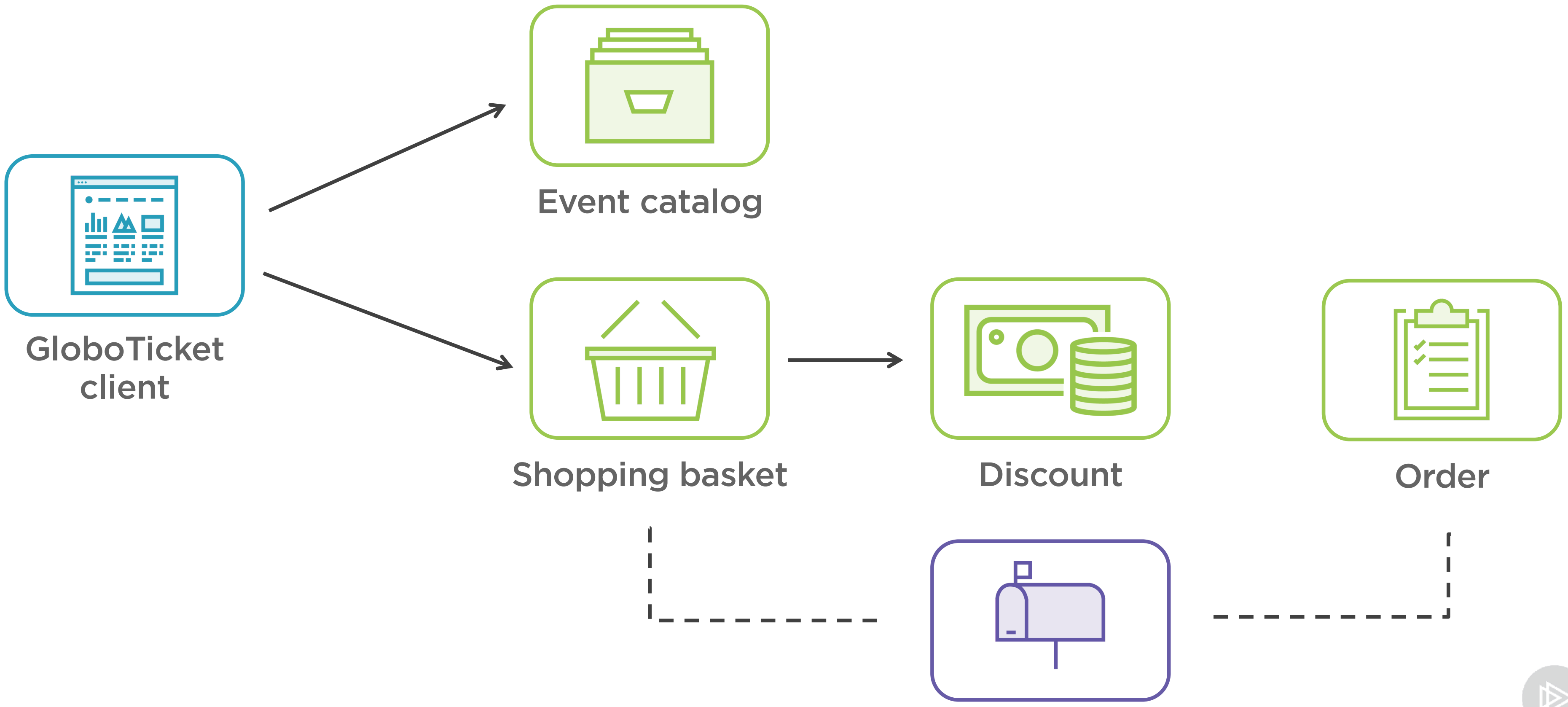
Introducing the Upcoming Demos



Introducing the Upcoming Demos



Introducing the Upcoming Demos



Introducing the Upcoming Demos

Request a token with “ordering” audience

- Token exchange flow

Include the token in each message

Pick up the message at the receiving end and validate the token to (dis)allow access

Demo

Requesting a token

Demo

Sending and validating a token

Dealing with Token Expiration

Validation will fail when the token has expired

Messages from the bus aren’t necessarily directly processed

```
var tokenValidationParameters = new TokenValidationParameters()
   {
       ValidAudience = "ordering",
       ValidIssuer = "https://localhost:5010",
       ValidateLifetime = false,
       IssuerSigningKeys = issuerSigningKeys
   };
```

Dealing with Token Expiration

Lifetime validation can be disabled

```
var tokenValidationParameters = new TokenValidationParameters()
   {
       ValidAudience = "ordering",
       ValidIssuer = "https://localhost:5010",
       ValidateLifetime = false,
       IssuerSigningKeys = issuerSigningKeys
   };
```

Dealing with Token Expiration

Lifetime validation can be disabled

Dealing with Token Expiration

Refresh tokens can be used to get new access tokens

- Send the refresh token together with the access token
- If the access token has expired, use the refresh token to get a new one

Dealing with Token Expiration

Complex

Secrets must be shared

Refresh tokens expire as well

Dealing with Token Expiration

We want to check whether the token allows us to process the message when the bus receives it

- Not when our code receives it

Check whether the token was expired at the moment the message was received by the bus

Demo

Dealing with token expiration

On application security…

You don’t need the most secure approach. You need the best fit for your application.

Choosing the Right Security Approach for Your Use Case

Don’t rely on external components to handle authentication / authorization

- Don’t trust the caller

Choosing the Right Security Approach for Your Use Case

- Sensitive data requires a tougher security approach
- Business requirements have an influence on the potential approach
- User experience has an influence on what’s feasible
- Performance issues should be taken into account
- Total cost of ownership (from development to keeping the application running) influences the security approach

Summary

Authenticating with the bus

- "Can an application access the bus?"
- Dependent on the type of bus

Granular authorization

- Build in on top of bus authentication
- Independent of the type of bus

Summary

Messages can stay on the bus for large amounts of time

- Validate whether the token allowed access when the message was received by the bus

@kevinDockx