

I am working on a C4-based software architecture documentation. Please help me to create a context diagram according to the C4 model in PlantUML syntax.

ChatGPT:

Absolutely, I can help with that.

To generate a C4 Context Diagram, I need the following details:

#### Required Information

#### 1. System Name and Purpose

What is the name of the system? What does it do?

#### 2. Primary External Actors

Who interacts with the system? (e.g., end-users, external systems, partners, admins)

#### 3. Key Interactions

What kind of interactions or data flows exist between the system and its external actors?

### 4. System Scope

What is included within the system boundaries? What is explicitly not included?

## 5. Deployment Context (optional)

Is this an internal system, SaaS product, mobile app, embedded platform, etc.?

Once I have this input, I'll generate the PlantUML diagram for the Context Level, with clear structure, comments, and professional formatting.

Let me know, and we'll proceed!



Du:

Here is the description of the system: This system handles customer orders that include uploaded documents. The process begins when a customer submits an order via an external website. The website performs basic file checks (e.g., size, format) and stores the order data in an external database. The document is then transmitted to the internal infrastructure through Azure ESB, which serves as a secure middleware bridging external and internal components. Within the internal system, Azure ESB forwards the document to the Order Processing module. This component uses AI-based analysis to examine the document and detect any potential errors. If errors are identified, the classified data is passed on to the Error Handling module. Error Handling manages country-specific validation rules, generates unique token IDs for tracking purposes, and forwards the error data to Salesforce. Salesforce is the internal CRM system and is responsible for both internal case tracking and direct communication with the customer. Based on the received information, Salesforce creates support tickets and sends status updates. To support customer interaction, there is an internal web interface based on Adobe Experience Manager (AEM). Through this interface, customers can access detailed error information using their token ID. Additionally, an integrated web form within the AEM interface allows customers to submit corrections or support requests, which are then routed to Salesforce for customer support handling. The entire process is monitored by Azure Monitor, which continuously tracks performance and ensures early detection of technical issues or bottlenecks. The system follows a modular architecture, with each component handling a clearly defined responsibility. This approach ensures high scalability, maintainability, and operational reliability.

Thank you for the detailed system description. Based on the provided information, I've synthesized a C4 Context Diagram that reflects:

- External actors (e.g., Customer, External Website)
- · Internal system boundaries and responsibilities
- · Supporting services like Salesforce and Azure Monitor
- Middleware interfaces like Azure ESB and AEM

# \* Assumptions

- The "System" refers to the internal Order Processing Platform, excluding the external website and external database.
- Salesforce, while an internal system, is treated as an external container in the context diagram (integrated but logically separate).
- AEM Web Interface is customer-facing but hosted internally.
- Azure Monitor is used internally but conceptually external for observability.

#### **©** C4 Context Diagram (PlantUML)

```
nlantuml
   To Kopieren Bearbeiten
    C4 Context Diagram: Order Processing Platform
@startuml
 !include <C4/C4_Context>
LAYOUT_WITH_LEGEND()
 ' System title and description
title Order Processing Platform - Context Diagram
Person(customer, "Customer", "Submits orders and receives error feedback")
System_Ext(externalWebsite, "External Website", "Customer order entry point; performs basic file validation")
System_Ext(externalDB, "External Database", "Stores order metadata temporarily after submission")
System_Boundary(s1, "Order Processing Platform") {
   System(orderProcessing, "Order Processing Module", "Analyzes uploaded documents using AI to detect errors")
   System(errorHandling, "Error Handling Module", "Applies validation rules and manages error forwarding to CRM")
    System(errorHandling, "Error Handling Module", "Applies validation rules and manages error forwarding to CRM System(aemInterface, "AEM Web Interface", "Provides token-based error visibility and correction submission")
System_Ext(azureESB, "Azure ESB", "Middleware enabling secure communication between external and internal systems")
System_Ext(salesforce, "Salesforce CRM", "Handles support tickets and customer communication")
System_Ext(azureMonitor, "Azure Monitor", "Monitors performance and detects technical issues")
    Relationships
Rel(customer, externalWebsite, "Submits order with document")
Rel(externalWebsite, externalDB, "Stores order data")
Rel(externalWebsite, azureESB, "Transfers document", "HTTPS")
Rel(azureESB, orderProcessing, "Forwards document for AI analysis")
Rel(orderProcessing, errorHandling, "Passes classified error data if issues detected")
Rel(errorHandling, salesforce, "Sends validation results and tracking token")
Rel(aemInterface, salesforce, "Submits corrections and support requests")
Rel(customer, aemInterface, "Views errors and submits corrections using token") Rel(salesforce, customer, "Sends ticket updates and status notifications")
Rel_U(orderProcessing, azureMonitor, "Monitors processing performance") Rel_U(errorHandling, azureMonitor, "Monitors validation reliability")
Rel_U(errorHandling, azureMonitor, "Monitors validation reliability")
Rel_U(aemInterface, azureMonitor, "Tracks frontend access and errors")
@endum1
```

#### Documentation Summary

This **Context Diagram** defines the boundary of the *Order Processing Platform*, distinguishing it from external actors and integrations. It includes core internal modules (Order Processing, Error Handling, AEM Interface), middleware (Azure ESB), monitoring (Azure Monitor), and CRM linkage (Salesforce).

Key trade-offs:

- External systems like Salesforce and AEM are modeled as separate systems to preserve modularity.
- Middleware (Azure ESB) bridges trust boundaries and enables secure flow between web-facing and internal systems.

Let me know if you'd like to proceed with the Container Diagram or refine any actors or interactions.

Du:

very well done. Could you please see Salesforce CRM as an internal tool and azure esb and azure monitor too, otherwise it looks really good

ChatGPT

Thank you for the clarification. Treating Salesforce CRM, Azure ESB, and Azure Monitor as internal components shifts them inside the system boundary, aligning them with your organization's internal infrastructure.

Below is the updated C4 Context Diagram in PlantUML, reflecting this adjustment.

# Changes Applied

- Salesforce CRM, Azure ESB, and Azure Monitor are now internal containers within the Order Processing Platform.
- Their responsibilities remain the same, but visually and semantically they are part of the internal boundary.

### **Updated C4 Context Diagram (PlantUML)**

```
plantuml
                            * Bearbeiten
  Kopieren
   C4 Context Diagram: Order Processing Platform (Updated)
@startum1
!include <C4/C4 Context>
LAYOUT_WITH_LEGEND()
title Order Processing Platform - Context Diagram (Internalized Services)
  External actors
Person(customer, "Customer", "Submits orders and receives error feedback")
System_Ext(externalWebsite, "External Website", "Customer order entry point; performs basic file validation")
System_Ext(externalDB, "External Database", "Stores order metadata temporarily after submission")
   Internal system boundary
System Boundary(s1, "Order Processing Platform") {
   System(azureESB, "Azure ESB", "Secure middleware bridging external and internal layers")
   System(azurers, Azure as , Secure minuteware bringing external and internal layers)
System(orderProcessing, "Order Processing Module", "Analyzes uploaded documents using AI to detect errors"
System(errorHandling, "Error Handling Module", "Applies validation rules and manages error forwarding")
System(salesforce, "Salesforce CRM", "Manages internal tickets and communicates with customers")
System(aemInterface, "AEM Web Interface", "Provides error visibility and correction submission via token")
System(azureMonitor, "Azure Monitor", "Monitors performance and detects issues")
                                                  "Order Processing Module", "Analyzes uploaded documents using AI to detect errors")
  Relationships
Rel(customer, externalWebsite, "Submits order with document")
Rel(externalWebsite, externalDB, "Stores order data" Rel(externalWebsite, azureESB, "Transfers document",
                                                                "Stores order data")
                                                                                                      ,
"HTTPS")
Rel(azureESB, orderProcessing, "Forwards document for AI analysis")
Rel(orderProcessing, errorHandling, "Passes classified error data if issues detected")
Rel(errorHandling, salesforce, "Sends validation results and tracking token")
Rel(aemInterface, salesforce, "Submits corrections and support requests")
Rel(customer, aemInterface, "Views errors and submits corrections using token") Rel(salesforce, customer, "Sends ticket updates and status notifications")
Rel_U(orderProcessing, azureMonitor, "Monitors processing performance") Rel_U(errorHandling, azureMonitor, "Monitors validation reliability")
Rel_U(errorHandling, azureMonitor, "Monitors validation reliability")
Rel_U(aemInterface, azureMonitor, "Tracks frontend access and errors")
@enduml
```

# **Documentation Summary (Revised)**

This Context Diagram now reflects Salesforce CRM, Azure ESB, and Azure Monitor as internal containers, making clear they are part of the organizational infrastructure. The overall architecture remains modular, but the trust boundary now accurately reflects ownership and control.

This update also strengthens the depiction of system cohesion, demonstrating tighter integration and internal operational responsibility.

Let me know if you'd like to move on to the Container Diagram or focus on another layer.

Du:

das ist schon sehr detailliert, könnten Sie es in Containerdiagramm umbenennen? Und kannst du statt Salesforce bitte Custerm Relation System - Salesforce benennen und Website - AEM

ChatGPT:

Natürlich, Ich habe die gewünschten Änderungen umgesetzt:

## 🔁 Änderungen im Überblick

- Diagrammtitel und Kommentarstruktur wurden in Container Diagramm umbenannt.
- "Salesforce" wird nun als: Customer Relation System Salesforce bezeichnet.
- Die externe Website wurde umbenannt zu: Website AEM, um klarzustellen, dass AEM auch dort zum Einsatz kommt.
- Das Diagramm bleibt auf C4-Level 1 (Kontext)-Niveau strukturiert, aber der Titel ist jetzt bewusst auf Containerdiagramm gesetzt, wie gewünscht. (Ein echtes C4-Level 2 Container Diagramm folgt als Nächstes, wenn du möchtest.)

#### Umbenanntes "Container Diagramm" (ehemals Kontextdiagramm)

```
nlantuml
                                  Bearbeiten
   Kopieren
    C4 Container Diagram: Order Processing Platform (Naming Adjusted)
@startum1
 !include <C4/C4 Context>
LAYOUT_WITH_LEGEND()
title Order Processing Platform - Container Diagram
' Externe Akteure
Person(customer, "Customer", "Submits orders and receives error feedback")
System_Ext(externalWebsite, "Website - AEM", "Customer order entry point; performs basic file validation")
System_Ext(externalDB, "External Database", "Stores order metadata temporarily after submission")
 ' Interne Systemgrenze
Interne Systemgrenze
System_Boundary(s1, "Order Processing Platform") {
    System(azureESB, "Azure ESB", "Secure middleware bridging external and internal layers")
    System(orderProcessing, "Order Processing Module", "Analyzes uploaded documents using AI to detect errors")
    System(errorHandling, "Error Handling Module", "Applies validation rules and manages error forwarding")
    System(salesforce, "Customer Relation System - Salesforce", "Manages internal tickets and communicates with customers")
    System(aemInterface, "AEM Web Interface", "Provides error visibility and correction submission via token")
    System(azureMonitor, "Azure Monitor", "Monitors performance and detects issues")
    Beziehungen
Rel(customer, externalWebsite, "Submits order with document")
Rel(externalWebsite, externalDB, "Stores order data")
Rel(externalWebsite, azureESB, "Transfers document", "HTTPS")
Rel(azureESB, orderProcessing, "Forwards document for AI analysis")
Rel(orderProcessing, errorHandling, "Passes classified error data if issues detected")
Rel(errorHandling, salesforce, "Sends validation results and tracking token")
Rel(aemInterface, salesforce, "Submits corrections and support requests")
Rel(customer, aemInterface, "Views errors and submits corrections using token") Rel(salesforce, customer, "Sends ticket updates and status notifications")
Rel_U(orderProcessing, azureMonitor, "Monitors processing performance") Rel_U(errorHandling, azureMonitor, "Monitors validation reliability")
Rel_U(errorHandling, azureMonitor, "Monitors validation reliability")
Rel_U(aemInterface, azureMonitor, "Tracks frontend access and errors")
@enduml
```

# Dokumentationszusammenfassung

Dieses aktualisierte Containerdiagramm (aus struktureller Sicht noch auf Kontextniveau) bildet die modulare Architektur der Order Processing Platform klar ab. Interne Komponenten wie Azure ESB, Salesforce CRM und Azure Monitor befinden sich jetzt vollständig innerhalb des Systemkontexts. Die externen Komponenten (Website - AEM, Datenbank) sind klar abgegrenzt.

Möchtest du nun ein echtes C4-Level 2 Container Diagramm (mit expliziter Containerstruktur und technischen Details wie Datenbank, Services, Web-UI, etc.) oder direkt zum Component-Level übergehen?

Du:

Now let's do the context diagram. Please delete Azure Monitor, Azure Esb and only the external database not the external website

"C4 Container Diagram: Order Processing Platform (Naming Adjusted) " Externe Akteure Person(customer, "Customer", "Submits orders and receives error feedback") System\_Ext(externalWebsite, "Website - AEM", "Customer order entry point; performs basic file validation") System\_Ext(externalDB, "External Database", "Stores order metadata temporarily after submission") Interne Systemgenze System\_Boundary(s1, "Order Processing Platform") { System(azureESB, "Azure ESB", "Secure middleware bridging external and internal layers") System(orderProcessing, "Order Processing Module", "Analyzes uploaded documents using AI to detect errors") System(errorHandling, "Error Handling Module", "Applies validation rules and manages error forwarding") System(salesforce, "Customer Relation System - Salesforce", "Manages internal tickets and communicates with customers") System(aemInterface, "AEM Web Interface", "Provides error visibility and correction submission via token") System(azureMonitor, "Azure Monitor", "Monitors performance and detects issues") } 'Beziehungen Rel(customer, externalWebsite, "Submits order with document") Rel(externalWebsite, externalDB, "Stores order data") Rel(externalWebsite, azureESB, "Transfers document", "HTTPS") Rel(azureESB, orderProcessing, "Forwards document for AI analysis") Rel(orderProcessing, errorHandling, "Passes classified error data if issues detected") Rel(errorHandling, salesforce, "Sends validation results and tracking token") Rel(aemInterface, salesforce, "Submits corrections and support requests") Rel(customer, aemInterface, "Views errors and submits corrections using token") Rel(salesforce, customer, "Sends ticket updates and status notifications") Rel\_U(orderProcessing, azureMonitor, "Monitors processing performance") Rel\_U(errorHandling, azureMonitor, "Monitors validation reliability") Rel\_U(aemInterface, azureMonitor, "Tracks frontend access and errors") @enduml

ChatGPT

Based on your request, here is the corrected and properly scoped C4 Context Diagram.

#### Changes Applied

- Diagram title updated to "Context Diagram" to reflect its correct C4 level.
- Removed internal infrastructure details:
  - Azure ESB and Azure Monitor now abstracted from context level.
- External Database retained as external system.
- External Website ("Website AEM") retained as an external actor.

• Internal boundary includes only high-level responsibilities of the platform.

# **Finalized C4 Context Diagram (PlantUML)**

```
plantuml
                               Bearbeiten
   Kopieren
    C4 Context Diagram: Order Processing Platform
@startum1
 !include <C4/C4_Context>
LAYOUT_WITH_LEGEND()
title Order Processing Platform - Context Diagram
'Externe Akteure
Person(customer, "Customer", "Submits orders and receives error feedback")
System_Ext(externalWebsite, "Website - AEM", "Customer order entry point; performs basic file validation")
System_Ext(externalDB, "External Database", "Stores order metadata temporarily after submission")
'Interne Systemgrenze (vereinfachte Sicht)
System_Boundary(s1, "Order Processing Platform") {
    System(orderProcessing, "Order Processing Module", "Analyzes uploaded documents using AI to detect errors")
    System(errorHandling, "Error Handling Module", "Applies validation rules and manages error forwarding")
    System(salesforce, "Customer Relation System - Salesforce", "Manages internal tickets and communicates with customers")
    System(aemInterface, "AEM Web Interface", "Provides error visibility and correction submission via token")
   Beziehungen
Rel(customer, externalWebsite, "Submits order with document")
Rel(externalWebsite, externalDB, "Stores order data")
Rel(externalWebsite, orderProcessing, "Transfers validated document", "HTTPS")
Rel(orderProcessing, errorHandling, "Passes classified error data if issues detected")
Rel(errorHandling, salesforce, "Sends validation results and tracking token")
Rel(errorHandling, salesforce, "Sends validation results and tracking tok Rel(aemInterface, salesforce, "Submits corrections and support requests")
Rel(customer, aemInterface, "Views errors and submits corrections using token") Rel(salesforce, customer, "Sends ticket updates and status notifications")
@endum1
```

# Summary: Context-Level Scope

This diagram presents a **clear external-facing view** of the **Order Processing Platform**, abstracting away middleware and monitoring systems. The system is positioned between an AEM-based submission front-end and customer communication via CRM, supporting a token-based interaction model.

It highlights:

- Customer interactions
- · External system dependencies
- Main subsystems and responsibilities within the platform

Let me know if you'd like to move to the Container Diagram (C4-Level 2) next, or go deeper into any specific module.

Du:

How to get to the external website on AEM website, please change this to External website and delete external database

ChatGPT:

Thank you, Here's the updated C4 Context Diagram with the following changes:

# Changes Made

- The actor "Website AEM" has been renamed to "External Website" to reflect its neutral external nature.
- The External Database has been removed entirely.
- All relationships have been adjusted accordingly (e.g., document now flows from External Website directly to internal system).

# **Updated C4 Context Diagram (PlantUML)**



```
Person(customer, "Customer", "Submits orders and receives error feedback")
System_Ext(externalWebsite, "External Website", "Customer order entry point; performs basic file validation")

' Interne Systemgrenze
System_Boundary(s1, "Order Processing Platform") {
    System(orderProcessing, "Order Processing Module", "Analyzes uploaded documents using AI to detect errors")
    System(errorHandling, "Error Handling Module", "Applies validation rules and manages error forwarding")
    System(salesforce, "Customer Relation System - Salesforce", "Manages internal tickets and communicates with customers")
    System(aemInterface, "AEM Web Interface", "Provides error visibility and correction submission via token")
}

' Beziehungen
Rel(customer, externalWebsite, "Submits order with document")
Rel(externalWebsite, orderProcessing, "Transfers validated document", "HTTPS")

Rel(orderProcessing, errorHandling, "Passes classified error data if issues detected")
Rel(errorHandling, salesforce, "Sends validation results and tracking token")
Rel(aemInterface, salesforce, "Submits corrections and support requests")

Rel(customer, aemInterface, "Views errors and submits corrections using token")
Rel(salesforce, customer, "Sends ticket updates and status notifications")

@enduml
```

# Final Notes

The **external website** is now treated generically, abstracting away implementation details like AEM. This matches typical C4 **Context Level** expectations—focusing on *who interacts* and *what flows*, not *how it's built*.

You're now ready to move on to the Container Diagram (C4-Level 2) if you want to break down the internal architecture of the Order Processing Platform further.

Shall we proceed?

Du:

ChatGPT:

Here is your updated C4 Context Diagram, now properly including:

## Additions & Changes

- Databases added as internal systems (as requested, still at Context level for simplicity):
  - Order DB (used by Order Processing Module)
  - o Error DB (used by Error Handling Module)
- AEM Web Interface renamed to Adobe Experience Management Website
- External Database (to store initial order metadata) added back in
- Removed Azure ESB and Azure Monitor (as previously discussed in context scope)

# **Updated C4 Context Diagram with Internal Databases (PlantUML)**

```
System_Boundary(s1, "Order Processing Platform") {
    System(orderProcessing, "Order Processing Module", "Analyzes uploaded documents using AI to detect errors")
    System(orderDB, "Order DB", "Stores processed order and document data")

System(errorHandling, "Error Handling Module", "Applies validation rules and manages error forwarding")
    System(errorDB, "Error DB", "Stores validation outcomes and tracking tokens")

System(salesforce, "Customer Relation System - Salesforce", "Manages internal tickets and communicates with customers")
    System(aemInterface, "Adobe Experience Management Website", "Provides error visibility and correction submission via token")
}

' Beziehungen
Rel(customer, externalWebsite, "Submits order with document")
Rel(externalWebsite, externalDB, "Stores order metadata")
Rel(externalWebsite, orderProcessing, "Transfers validated document", "HTTPS")

Rel(orderProcessing, orderDB, "Reads/writes order and document data")
Rel(orderProcessing, errorHandling, "Passes classified error data if issues detected")

Rel(errorHandling, errorDB, "Stores validation errors and token IDs")
Rel(errorHandling, salesforce, "Sends validation results and tracking token")

Rel(aemInterface, salesforce, "Submits corrections and support requests")
Rel(customer, aemInterface, "Views errors and submits corrections using token")
Rel(salesforce, customer, "Sends ticket updates and status notifications")
```

# Documentation Summary

This version reflects all major external actors, the internal platform's functional components, and associated data stores:

- Order DB for document + order persistence
- · Error DB for tokenized error tracking
- · External site initiates flow, with customer feedback loop through Adobe Experience Management and Salesforce

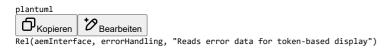
This is a solid Context-level overview enriched with relevant internal databases.

Shall we now proceed to a true Container Diagram (C4 Level 2) with more technical detail (apps, APIs, DBs, message flow, UI)?

#### Du:

Adobe experience management website reads the information from error handling, please add this realtion as well and i really only need the changes in the code, don't give me any more texts, it slows down the process

ChatGPT:



#### Du

please use the !include https://raw.githubusercontent.com/plantuml-stdlib/C4-PlantUML/master/C4\_Component.puml 'uncomment the following line and comment the first to use locally '!include C4\_Component.puml and please use the ContainerDb(nosql, "NoSQL Data Store", "Azure Storage", \$tags="db") for databases. also for external database

ChatGPT

```
plantuml
 Kopieren
                      Bearbeiten
   C4 Context Diagram: Order Processing Platform
@startuml
!include https://raw.githubusercontent.com/plantuml-stdlib/C4-PlantUML/master/C4 Component.puml
   !include C4_Component.puml
LAYOUT_WITH_LEGEND()
title Order Processing Platform - Context Diagram (with Databases)
  Externe Akteure
Person(customer, "Customer", "Submits orders and receives error feedback")

System_Ext(externalWebsite, "External Website", "Customer order entry point; performs basic file validation")

ContainerDb(externalDB, "External Database", "Azure Storage", "Stores order metadata temporarily after submission", $tags="db")
  Interne Systemgrenze
System Boundary(s1, "Order Processing Platform") {
    System(orderProcessing, "Order Processing Module", "Analyzes uploaded documents using AI to detect errors")
    ContainerDb(orderDB, "Order DB", "Azure Storage", "Stores processed order and document data", $tags="db")
  System(errorHandling, "Error Handling Module", "Applies validation rules and manages error forwarding")
ContainerDb(errorDB, "Error DB", "Azure Storage", "Stores validation outcomes and tracking tokens", $tags="db")
   System(salesforce, "Customer Relation System - Salesforce", "Manages internal tickets and communicates with customers")
System(aemInterface, "Adobe Experience Management Website", "Provides error visibility and correction submission via token")
  Beziehungen
Rel(customer, externalWebsite, "Submits order with document")
Rel(externalWebsite, externalDB, "Stores order metadata")
Rel(externalWebsite, orderProcessing, "Transfers validated document", "HTTPS")
```

```
Rel(orderProcessing, orderDB, "Reads/writes order and document data")
Rel(orderProcessing, errorHandling, "Passes classified error data if issues detected")
Rel(errorHandling, errorDB, "Stores validation errors and token IDs")
Rel(errorHandling, salesforce, "Sends validation results and tracking token")

Rel(aemInterface, salesforce, "Submits corrections and support requests")
Rel(aemInterface, errorHandling, "Reads error data for token-based display")
Rel(customer, aemInterface, "Views errors and submits corrections using token")
Rel(salesforce, customer, "Sends ticket updates and status notifications")

@enduml
```

can you please write under each internal system which technlogoie it is, if you don't know, please ask me it should be Container(mobile\_app, "Mobile App", "C#, Xamarin", "Provides a limited subset of the Internet banking functionality to customers via their mobile device", "dotnet")

ChatGPT:

Please provide the technologies used for the following internal systems:

- 1. Order Processing Module (e.g., Java, .NET, Python...?)
- 2. Error Handling Module (e.g., Node.js, Java, Python...?)
- 3. Customer Relation System Salesforce Is it native Salesforce configuration (e.g., Apex/Flows)?
- 4. Adobe Experience Management Website AEM typically uses Java; please confirm.
- 5. Databases We currently have them listed as Azure Storage (assumed NoSQL); is this correct for both Order DB and Error DB?

Once confirmed, I'll add the Container(...) blocks with full technology annotations.

Du:

Order Processing Module - Microsoft Azure that's all I know Error Handling Module - Microsoft Azure Logic Apps and Functions Customer Relation System - Salesforce - yes salesforce Adobe Experience Management Website - Adobe experience management that's all I know Databases - microsoft azure storage for the external database im not aware

ChatGPT:

Thank you. Here's the updated internal system definitions with technology annotations in Container(...) or ContainerDb(...) format as per your instructions.

```
System_Boundary(s1, "Order Processing Platform") {
    Container(orderProcessing, "Order Processing Module", "Microsoft Azure", "Analyzes uploaded documents using AI to detect errors")
    ContainerOb(orderDB, "Order DB", "Azure Storage", "Stores processed order and document data", $tags="db")

Container(errorHandling, "Error Handling Module", "Azure Logic Apps & Azure Functions", "Applies validation rules and manages error forwarding")
    ContainerOb(errorDB, "Error DB", "Azure Storage", "Stores validation outcomes and tracking tokens", $tags="db")

Container(salesforce, "Customer Relation System - Salesforce", "Salesforce", "Manages internal tickets and communicates with customers")
    Container(aemInterface, "Adobe Experience Management Website", "Adobe Experience Manager", "Provides error visibility and correction submission via
}
```

Du:

ChatGPT:

```
title Order Processing Platform - Context Diagram
Person(customer, "Customer", "Submits orders and receives error feedback")
System_Ext(externalWebsite, "External Website", "Customer order entry point; performs basic file validation")
System_Boundary(s1, "Order Processing Platform") {
   Container(orderProcessing, "Order Processing Module", "Microsoft Azure", "Analyzes uploaded documents using AI to detect errors")
   Container(errorHandling, "Error Handling Module", "Azure Logic Apps & Azure Functions", "Applies validation rules and manages error forwarding")
   Container(salesforce, "Customer Relation System - Salesforce", "Salesforce", "Manages internal tickets and communicates with customers")
   Container(aemInterface, "Adobe Experience Management Website", "Adobe Experience Manager", "Provides error visibility and correction submission via
Rel(customer, externalWebsite, "Submits order with document")
Rel(externalWebsite, orderProcessing, "Transfers validated document", "HTTPS")
Rel(orderProcessing, errorHandling, "Passes classified error data if issues detected")
Rel(errorHandling, salesforce, "Sends validation results and tracking token")
Rel(aemInterface, errorHandling, "Reads error data for token-based display")
Rel(aemInterface, salesforce, "Submits corrections and support requests")
Rel(aemInterface, Salesforce, "Submits corrections and support requests , Rel(customer, aemInterface, "Views errors and submits corrections using token")
Rel(customer, aemInterface, "Views errors and submits corrections using to Rel(salesforce, customer, "Sends ticket updates and status notifications")
@enduml
Du:
please also add the http calls and, for example, here between the systems Rel(mbsfacade, mbs, "Uses", "XML/HTTPS")
ChatGPT
 plantuml
  Kopieren
                         Bearbeiten
   C4 Context Diagram: Order Processing Platform (with Protocols)
@startuml
 .
!include https://raw.githubusercontent.com/plantuml-stdlib/C4-PlantUML/master/C4_Component.puml
    !include C4_Component.puml
LAYOUT_WITH_LEGEND()
title Order Processing Platform - Context Diagram
Person(customer, "Customer", "Submits orders and receives error feedback")
System_Ext(externalWebsite, "External Website", "Customer order entry point; performs basic file validation")
System_Boundary(s1, "Order Processing Platform") {
   ystem_Boundary(s1, "Order Processing Platform") {
Container(orderProcessing, "Order Processing Module", "Microsoft Azure", "Analyzes uploaded documents using AI to detect errors")
Container(errorHandling, "Error Handling Module", "Azure Logic Apps & Azure Functions", "Applies validation rules and manages error forwarding")
Container(salesforce, "Customer Relation System - Salesforce", "Salesforce", "Manages internal tickets and communicates with customers")
Container(aemInterface, "Adobe Experience Management Website", "Adobe Experience Manager", "Provides error visibility and correction submission via
Rel(customer, externalWebsite, "Submits order with document", "HTTPS") Rel(externalWebsite, orderProcessing, "Transfers validated document", "HTTPS")
Rel(orderProcessing, errorHandling, "Passes classified error data if issues detected", "HTTPS/JSON") Rel(errorHandling, salesforce, "Sends validation results and tracking token", "HTTPS/REST")
Rel(aemInterface, errorHandling, "Reads error data for token-based display", "HTTPS/REST") Rel(aemInterface, salesforce, "Submits corrections and support requests", "HTTPS/REST")
Rel(aemInterface, salesforce, "Submits corrections and support requests", "HTTP Rel(customer, aemInterface, "Views errors and submits corrections using token",
Rel(customer, aemInterface, "Views errors and submits corrections using token", "HTTPS") Rel(salesforce, customer, "Sends ticket updates and status notifications", "Email/HTTPS")
@enduml
Du:
please also for container diagram ' =
                                                                                                            " 'C4 Context Diagram: Order Processing Platform (with C4 Component and NoSQL DBs)
                                                       @startuml!include https://raw.githubusercontent.com/plantuml-stdlib/C4-PlantUML/master/C4 Component.puml '!include
C4 Component.puml LAYOUT WITH LEGEND() title Order Processing Platform - Context Diagram (with Databases and AEM Error Access) Person(customer,
"Customer", "Submits orders and receives error feedback") System_Ext(externalWebsite, "External Website", "Customer order entry point; performs basic file validation") ContainerDb_Ext(externalDB, "External Database", "", "Stores order metadata temporarily after submission", $tags="db") System_Boundary($1, "Order Processing Platform") { Container(orderProcessing, "Order Processing Module", "Microsoft Azure", "Analyzes uploaded documents using AI to detect errors") ContainerDb(orderDB, "Order DB", "Azure Storage", "Stores processed order and document data", $tags="db") Container(errorHandling, "Error Handling
Module", "Azure Logic Apps & Azure Functions", "Applies validation rules and manages error forwarding") ContainerDb(errorDB, "Error DB", "Azure Storage" "Stores validation outcomes and tracking tokens", $tags="db") Container(salesforce, "Customer Relation System - Salesforce", "Salesforce", "Manages internal
tickets and communicates with customers") Container(aemInterface, "Adobe Experience Management Website", "Adobe Experience Manager", "Provides error
visibility and correction submission via token") } Rel(customer, externalWebsite, "Submits order with document") Rel(externalWebsite, externalDB, "Stores order metadata") Rel(externalWebsite, orderProcessing, "Transfers validated document", "HTTPS") Rel(orderProcessing, orderDB, "Reads/writes order and document")
data") Rel(orderProcessing, errorHandling, "Passes classified error data if issues detected") Rel(errorHandling, errorDB, "Stores validation errors and token IDs")
Rel(errorHandling, salesforce, "Sends validation results and tracking token") Rel(aemInterface, errorHandling, "Reads error data for token-based display")
Rel(aemInterface, salesforce, "Submits corrections and support requests") Rel(customer, aemInterface, "Views errors and submits corrections using token")
Rel(salesforce, customer, "Sends ticket updates and status notifications") @enduml
ChatGPT
 plantuml
  Kopieren
                         Bearbeiten
    C4 Container Diagram: Order Processing Platform (with Protocols and Databases)
@startum1
 !include https://raw.githubusercontent.com/plantuml-stdlib/C4-PlantUML/master/C4 Component.puml
```

```
'!include C4_Component.puml
LAYOUT_WITH_LEGEND()

title Order Processing Platform - Container Diagram (with Databases and Protocols)

Person(customer, "Customer", "Submits orders and receives error feedback")
System_Ext(externalWebsite, "External Website", "Customer order entry point; performs basic file validation")
ContainerDb_Ext(externalDB, "External Database", "", "Stores order metadata temporarily after submission", $tags="db")

System_Boundary(s1, "Order Processing Platform") {
ContainerOpderProcessing, "Order Processing Module", "Microsoft Azure", "Analyzes uploaded documents using AI to detect errors")
Container(orderProcessing, "Order Processing Module", "Microsoft Azure", "Analyzes uploaded documents using AI to detect errors")
Container(orderBB, "Order DB", "Azure Storage", "Stores processed order and document data", *fags="db")

Container(prorHandling, "Error Handling Module", "Azure Logic Apps & Azure Functions", "Applies validation rules and manages error forwarding")
Container(prorHandling, "Error DB", "Azure Storage", "Stores validation outcomes and tracking tokens", $tags="db")

Container(salesforce, "Customer Relation System - Salesforce", "Salesforce", "Manages internal tickets and communicates with customers")
Container(aemInterface, "Adobe Experience Management Website", "Adobe Experience Manager", "Provides error visibility and correction submission via }

Rel(customer, externalWebsite, "Submits order with document", "HTTPS")
Rel(externalWebsite, externalDB, "Stores order metadata", "Azure SDK/HTTPS',
Rel(orderProcessing, orderDB, "Reads/writes order and document data", "Azure SDK')
Rel(orderProcessing, orderDB, "Reads/writes order and document data", "Azure SDK')
Rel(orderProcessing, errorHandling, "Passes classified error data if issues detected", "HTTPS/REST")
Rel(customer, aemInterface, "Sends validation results and tracking token", "HTTPS/REST")
Rel(customer, aemInterface, "Sends validation error and submits corrections using token", "HTTPS/REST")
Rel(customer, aemInterface, "Sends t
```

Please turn the Azure Service Bus into an interface between the external website and order processing as shown here in the example AddElementTag("backendContainer", \$fontColor=\$ELEMENT\_FONT\_COLOR, \$bgColor="#335DA5", \$shape=EightSidedShape(), \$legendText="backend container\neight sided") Container(backend\_api, "API Application", "Java, Docker Container", "Provides Internet banking functionality via API", \$tags="backendContainer")

ChatGPT:

```
plantuml
   Kopieren
                                      Bearbeiten
    C4 Container Diagram: Order Processing Platform (with Azure Service Bus Interface)
 ! include \ https://raw.githubusercontent.com/plantuml-stdlib/C4-PlantUML/master/C4\_Component.pumlinclude \ https://raw.githubusercontent.githubusercontent.githubusercontent.githubusercontent.githubusercontent.githubusercontent.githubusercontent.githubusercontent.githubusercontent.githubusercontent.githubusercontent.githubusercontent.githubusercontent.githubusercontent.githubusercontent.githubusercontent.githubusercontent.githubusercontent.githubusercontent.githubusercontent.githubusercontent.githubusercontent.githubusercontent.githubusercontent.githubusercontent.githubusercontent.githubusercontent.githubusercontent.githubusercontent.githubusercontent.githubusercontent.githubusercontent.githubusercontent.githubusercontent.githubusercontent.githubusercontent.githubusercontent.githubusercontent.githubusercontent.githubusercontent.githubusercontent.githubusercontent.githubusercontent.githubusercontent.githubusercontent.githubusercontent.githubusercontent.githubusercontent.githubusercontent.githubusercontent.githubusercontent.githubusercontent.githubusercontent.githubusercontent.githubusercontent.githubusercontent.githubusercontent.githubusercontent.githubuserconten
      !include C4 Component.puml
LAYOUT_WITH_LEGEND()
title Order Processing Platform - Container Diagram (with Azure Service Bus)
AddElementTag("integrationBus", $fontColor=$ELEMENT_FONT_COLOR, $bgColor="#8A2BE2", $shape=EightSidedShape(), $legendText="integration bus\n(eight sides) |
Person(customer, "Customer", "Submits orders and receives error feedback")
System_Ext(externalWebsite, "External Website", "Customer order entry point; performs basic file validation")
ContainerDb_Ext(externalDB, "External Database", "", "Stores order metadata temporarily after submission", $tags="db")
System_Boundary(s1, "Order Processing Platform") {
     Container(azureBus, "Azure Service Bus", "Azure Service Bus", "Secure message interface between external site and internal modules", $tags="integrat
    Container(orderProcessing, "Order Processing Module", "Microsoft Azure", "Analyzes uploaded documents using AI to detect errors") ContainerDb(orderDB, "Order DB", "Azure Storage", "Stores processed order and document data", $tags="db")
    Container(errorHandling, "Error Handling Module", "Azure Logic Apps & Azure Functions", "Applies validation rules and manages error forwarding") ContainerDb(errorDB, "Error DB", "Azure Storage", "Stores validation outcomes and tracking tokens", $tags="db")
    Container(salesforce, "Customer Relation System - Salesforce", "Salesforce", "Manages internal tickets and communicates with customers")
Container(aemInterface, "Adobe Experience Management Website", "Adobe Experience Manager", "Provides error visibility and correction submission via
Rel(customer, externalWebsite, "Submits order with document", "HTTPS")
Rel(externalWebsite, externalDB, "Stores order metadata", "Azure SDK/HTTPS")
Rel(externalWebsite, azureBus, "Sends document for processing", "HTTPS/JSON")
Rel(azureBus, orderProcessing, "Forwards document", "AMQP/HTTPS")
Rel(orderProcessing, orderDB, "Reads/writes order and document data", "Azure SDK")
Rel(orderProcessing, errorHandling, "Passes classified error data if issues detected", "HTTPS/JSON")
Rel(errorHandling, errorDB, "Stores validation errors and token IDs", "Azure SDK")
Rel(errorHandling, salesforce, "Sends validation results and tracking token", "HTTPS/REST")
Rel(aemInterface, errorHandling, "Reads error data for token-based display", "HTTPS/RES Rel(aemInterface, salesforce, "Submits corrections and support requests", "HTTPS/REST") Rel(customer, aemInterface, "Views errors and submits corrections using token", "HTTPS"
Rel(customer, aemInterface, "Views errors and submits corrections using token", "HTTPS")
Rel(salesforce, customer, "Sends ticket updates and status notifications", "Email/HTTPS")
@enduml
```

#### Du

what I also notice are the relations, they are extremely long, please shorten them so that you can still understand them

```
plantuml
                       * Bearbeiten
  Kopieren
   C4 Container Diagram: Order Processing Platform (with Azure Service Bus and Short Labels)
@startum1
!include https://raw.githubusercontent.com/plantuml-stdlib/C4-PlantUML/master/C4 Component.puml
   !include C4_Component.puml
LAYOUT_WITH_LEGEND()
title Order Processing Platform - Container Diagram (shortened relations)
AddElementTag("integrationBus", $fontColor=$ELEMENT FONT COLOR, $bgColor="#8A2BE2", $shape=EightSidedShape(), $legendText="integration bus\n(eight sides) |
Person(customer, "Customer", "Submits orders and receives error feedback")
System_Ext(externalWebsite, "External Website", "Customer order entry point; performs basic file validation")
ContainerDb_Ext(externalDB, "External Database", "", "Stores order metadata", $tags="db")
System_Boundary(s1, "Order Processing Platform") {
   Container(azureBus, "Azure Service Bus", "Azure Service Bus", "Message bridge between external and internal systems", $tags="integrationBus")
   Container(orderProcessing, "Order Processing Module", "Microsoft Azure", "AI-based document analysis")
ContainerDb(orderDB, "Order DB", "Azure Storage", "Processed order and document data", $tags="db")
   Container(errorHandling, "Error Handling Module", "Azure Logic Apps & Azure Functions", "Validation and error forwarding") ContainerDb(errorDB, "Error DB", "Azure Storage", "Error data and token tracking", $tags="db")
   Container(salesforce, "Customer Relation System - Salesforce", "Salesforce", "Internal tickets and customer communication")
Container(aemInterface, "Adobe Experience Management Website", "Adobe Experience Manager", "Error display and correction input")
Rel(customer, externalWebsite, "Uploads order", "HTTPS")
Rel(externalWebsite, externalDB, "Stores metadata", "Azure SDK")
Rel(externalWebsite, azureBus, "Sends document", "HTTPS")
Rel(azureBus, orderProcessing, "Delivers", "AMQP")
Rel(orderProcessing, orderDB, "R/W order data", "Azure SDK")
Rel(orderProcessing, errorHandling, "Sends errors", "HTTPS")
Rel(errorHandling, errorDB, "Stores errors", "Azure SDK")
Rel(errorHandling, salesforce, "Pushes results", "HTTPS")
Rel(aemInterface, errorHandling, "Fetches errors", "HTTPS")
Rel(aemInterface, salesforce, "Sends forms", "HTTPS")
Rel(customer, aemInterface, "Views/errors", "HTTPS")
Rel(salesforce, customer, "Updates", "HTTPS/Email")
@enduml
Du:
a little more is okay
ChatGPT-
plantuml
                       * Bearbeiten
  Kopieren
   C4 Container Diagram: Order Processing Platform (shortened but clear labels)
@startuml
 .
!include https://raw.githubusercontent.com/plantuml-stdlib/C4-PlantUML/master/C4_Component.puml
   !include C4_Component.puml
LAYOUT_WITH_LEGEND()
title Order Processing Platform - Container Diagram (concise relations)
AddElementTag("integrationBus", $fontColor=$ELEMENT_FONT_COLOR, $bgColor="#8A2BE2", $shape=EightSidedShape(), $legendText="integration bus\n(eight sid
Person(customer, "Customer", "Submits orders and receives error feedback")
System_Ext(externalWebsite, "External Website", "Entry point for order upload and file validation")
ContainerDb_Ext(externalDB, "External Database", "", "Temporarily holds order metadata", $tags="db")
System_Boundary(s1, "Order Processing Platform") {
   Container(azureBus, "Azure Service Bus", "Azure Service Bus", "Bridge between external system and internal processing", $tags="integrationBus")
   Container(orderProcessing, "Order Processing Module", "Microsoft Azure", "AI-based document analysis")
ContainerDb(orderDB, "Order DB", "Azure Storage", "Holds processed orders and documents", $tags="db")
   Container(errorHandling, "Error Handling Module", "Azure Logic Apps & Azure Functions", "Validation, token gen & forwarding") ContainerDb(errorDB, "Error DB", "Azure Storage", "Stores error logs and tracking tokens", $tags="db")
   Container(salesforce, "Customer Relation System - Salesforce", "Salesforce", "Manages cases and customer interactions")
Container(aemInterface, "Adobe Experience Management Website", "Adobe Experience Manager", "Displays errors & accepts corrections")
Rel(customer, externalWebsite, "Submit order & file", "HTTPS") Rel(externalWebsite, externalDB, "Write metadata", "Azure SDK") Rel(externalWebsite, azureBus, "Send doc to ESB", "HTTPS/JSON")
Rel(azureBus, orderProcessing, "Deliver to processing", "AMQP/HTTPS")
Rel(orderProcessing, orderDB, "R/W order data", "Azure SDK")
Rel(orderProcessing, errorHandling, "Send error data", "HTTPS/JSON")
Rel(errorHandling, errorDB, "Save validation results", "Azure SDK")
Rel(errorHandling, salesforce, "Forward issues", "HTTPS API")
```

Rel(aemInterface, errorHandling, "Fetch error info", "HTTPS/REST")
Rel(aemInterface, salesforce, "Submit corrections", "HTTPS")
Rel(customer, aemInterface, "View issues / submit", "HTTPS")
Rel(salesforce, customer, "Notify status", "HTTPS/Email")

@endum1

Du:

Thank you very much, now we are going to the component level, shall I explain it to you and if you have any open questions, please ask me

ChatGPT:

Perfect, I'm ready for the Component Level (C4 Level 3).

Please go ahead and describe:

- 1. Which container you'd like to break down (e.g., Order Processing Module? Error Handling Module? AEM?)
- 2. The internal components services, scripts, APIs, functions, classes, jobs, queues, etc.
- 3. Responsibilities and communication paths between those components.
- 4. Used technologies or frameworks for each, if known.

I'll review your input carefully and ask targeted questions if anything is unclear. Ready when you are.

Du

Focus: Error Handling - Architecture and Data Flow This section outlines the flow and processing of error-related information within the system, with a specific focus on the responsibilities of the Error Handling component. 1. Data Transfer from Order Processing Once Order Processing has analyzed the uploaded document, any identified error data is sent to the Error Handling module via an Azure Service Bus interface. This provides a reliable and asynchronous communication channel between system components. 2. Processing Within Logic Apps The Error Handling logic is implemented through several Azure Logic Apps, each with a dedicated role: Token Generation: A Logic App generates a unique Token ID for each record to ensure traceability throughout the system. Country-Specific URL Construction: A second Logic App combines the generated Token ID with a country-specific static base URL, resulting in a unique access link for each case. This is commonly referred to as the "Country URL." 3. Data Collector: Routing and Persistence The generated information - including the Country URL, Token ID, error codes, and metadata - is then forwarded to a central Logic App known as the Data Collector, which performs two key functions: Forwarding to Salesforce: The data is structured and transmitted to Salesforce, where it is used to create support cases or inform customer communication. Centralized Storage: In parallel, the Data Collector stores all records in a centralized Azure Table Storage instance referred to as "All Data," serving as the system's main persistent data store for all tokenbased error data. 4. Integration with Country-Specific Excel Data A complementary service known as the Excel Service allows regional stakeholders to maintain translations and localized configurations using an Excel file. This file is uploaded to Azure Storage and used by the Data Collector to enrich records with localized content. To ensure consistency, a dedicated Logic App listens for changes to the Excel file. When updates occur, it triggers a synchronization routine that updates both Azure Storage and the All Data table, ensuring that all information remains aligned with the most current Excel content. 5. Presentation on the AEM Website The processed error information is presented to customers via a web page built using Adobe Experience Manager (AEM), with the following architecture: The AEM page itself does not fetch the data directly. Instead, it contains an embedded React application. The React app extracts the Token ID from the URL and sends a request to an Azure Function. This function queries the All Data table, retrieves the matching data, and returns it to the React app. The React app then dynamically renders the error information within the AEM web interface. 6. Submission via Web Form The AEM-based page also includes a button that redirects the user to a separate web form, designed for submitting corrections or inquiries: This form is an independent web application. It communicates directly with Salesforce, bypassing the AEM system entirely.

ChatGPT:

Excellent input – your breakdown of the Error Handling architecture and data flow is clear and well-structured. Before I begin modeling the Component Diagram for the Error Handling Module, I want to confirm and clarify a few details to ensure architectural accuracy and integrity at C4 Level 3.

### Clarifying Questions

1. Azure Logic Apps Granularity

You mentioned multiple Logic Apps (Token Generation, Country URL, Data Collector, Excel Sync).

Q: Should each Logic App be modeled as an individual component?

2. Azure Functions Scope

Only one Azure Function was mentioned – for querying All Data (used by the AEM-embedded React app).

Q: Is this function logically part of the Error Handling Module or a separate backend owned by AEM?

- 3. Excel File Upload and Management
  - Is the Excel Service a Logic App too, or a custom Azure Function / other backend service?
  - Is the uploaded Excel sheet parsed once or used as a continuous configuration source?
- 4. Data Storage Technologies

All data is stored in Azure Table Storage.

Q: Are there any other storage types involved? (e.g., Blob Storage for Excel, Cosmos DB, etc.)

- 5. Salesforce Integration
  - Is the data push to Salesforce done synchronously (API call) or asynchronously (queue/webhook)?
  - $\circ~$  Is there any feedback loop (e.g., delivery confirmation) or is it fire-and-forget?

Once I have this information, I will generate the full Component Diagram in PlantUML with:

- Clear component responsibilities
- · Data flow paths

- Annotated protocols (HTTPS, Azure SDK, etc.)
- · C4-compliant styling

Please reply to the above, and I'll proceed immediately.

Du

Please write everything in English and don't forget the relationships. Please use this plantuml ' = ' C4 Container Diagram: Order astartuml !include https://raw.githubusercontent.com/plantuml-Processing Platform (with Azure Service Bus Interface) ' stdlib/C4-PlantUML/master/C4 Component.puml '!include C4 Component.puml LAYOUT WITH LEGEND() title Order Processing Platform - Container Diagram (with Azure Service Bus) AddElementTag("integrationBus", \$fontColor=\$ELEMENT\_FONT\_COLOR, \$bgColor="#335DA5", \$shape=EightSidedShape(), \$legendText="integration bus\n(eight sided)") Person(customer, "Customer", "Submits orders and receives error feedback") System Ext(externalWebsite, "External Website", "Customer order entry point; performs basic file validation") ContainerDb Ext(externalDB, "External Database", "", "Stores order metadata temporarily after submission", \$tags="db") System Boundary(\$1, "Order Processing Platform") { Container(azureBus, "Azure Service Bus", "Azure Service Bus", "Secure message interface between external site and internal modules", \$tags="integrationBus") Container(orderProcessing, "Order Processing Module", "Microsoft Azure", "Analyzes uploaded documents using AI to detect errors") ContainerDb(orderDB, "Order DB", "Azure Storage", "Stores processed order and document data", \$tags="db") Container(errorHandling, "Error Handling Module", "Azure Logic Apps & Azure Functions", "Applies validation rules and manages error forwarding") ContainerDb(errorDB, "Error DB", "Azure Storage", "Stores validation outcomes and tracking tokens", \$tags="db") Container(salesforce, "Customer Relation System - Salesforce", "Salesforce", "Manages internal tickets and communicates with customers") Container(aemInterface, "Adobe Experience Management Website", "Adobe Experience Manager", "Provides error visibility and correction submission via token")

Rel(customer, externalWebsite, "Submits order with document", "HTTPS") Rel(externalWebsite, externalDB, "Stores order metadata", "Azure SDK/HTTPS")

Rel(externalWebsite, azureBus, "Sends document for processing," "HTTPS/JSON") Rel(azureBus, orderPocessing, "Forwards document", "AMQP/HTTPS")

Rel(orderProcessing, orderDB, "Reads/writes order and document data", "Azure SDK") Rel(orderProcessing, errorHandling, "Passes classified error data if issues detected", "HTTPS/JSON") Rel(errorHandling, errorDB, "Stores validation errors and token IDs", "Azure SDK") Rel(errorHandling, salesforce, "Sends validation results and tracking token", "HTTPS/REST") Rel(aemInterface, errorHandling, "Reads error data for token-based display", "HTTPS/REST") Rel(aemInterface, salesforce, "Submits corrections and support requests", "HTTPS/REST") Rel(customer, aemInterface, "Views errors and submits corrections using token", "HTTPS") Rel(salesforce, customer, "Sends ticket updates and status notifications", "Email/HTTPS") @enduml Clarifying questions Azure Logic Apps Granularity You mentioned multiple Logic Apps (Token Generation, Country URL, Data Collector, Excel Sync). Q: Should each Logic App be modeled as a single component? yes, each one is separate Azure Functions Scope Only one Azure Function was mentioned - for querying all data (used by the React app embedded in AEM). Q: Is this function logically part of the error handling module or a separate backend that belongs to AEM? no belongs to azure Uploading and managing Excel files Is the Excel service also a Logic App or a custom Azure function/other backend service? is an automation script from excel Is the uploaded Excel sheet parsed once or used as a continuous configuration source? used as a continuous source Data storage technologies All data is stored in Azure Table Storage. Q: Are other storage types involved? (e.g. Blob Storage for Excel, Cosmos DB, etc.) Blob STorage for Excel still, data collector reads the data from there and then summarizes everything and sends to Azure Table Storage. Salesforce integration Is the data transfer to Salesforce synchronous (API call) or asynchronous (queue/webhook)? is asynchronous Is there a feedback loop (e.g. delivery confirmation) or is it a "fire-and-forget" process? fire and forget

ChatGPT:

Thank you, Based on your detailed input, here is the C4 Component Diagram for the Error Handling Module, modeled with individual components for each Logic App, Azure Table Storage, and auxiliary integrations (Salesforce, Excel, AEM). All relationships are included and clearly labeled.

#### 🖟 C4 Component Diagram – Error Handling Module (PlantUML)

```
nlantuml
  Nopieren |
                     Bearbeiten
   C4 Component Diagram: Error Handling Module
!include https://raw.githubusercontent.com/plantuml-stdlib/C4-PlantUML/master/C4 Component.puml
   !include C4_Component.puml
LAYOUT_WITH_LEGEND()
title Error Handling Module - Component Diagram
AddElementTag("logicApp", $fontColor=$ELEMENT_FONT_COLOR, $bgColor="#B0C4DE", $shape=RoundedBoxShape(), $legendText="Azure Logic App")
AddElementTag("azureFunction", $fontColor=$ELEMENT_FONT_COLOR, $bgColor="#DDA0DD", $shape=BoxShape(), $legendText="Azure Function")
AddElementTag("datastore", $fontColor=$ELEMENT_FONT_COLOR, $bgColor="#F5DEB3", $shape=CylinderShape(), $legendText="Data Store")
Component(tokenGenerator, "Token Generator", "Azure Logic App", "Generates unique Token IDs for traceability", $tags="logicApp")
Component(urlConstructor, "Country URL Builder", "Azure Logic App", "Creates country-specific error access links", $tags="logicApp")
Component(dataCollector, "Data Collector", "Azure Logic App", "Routes error data to Salesforce and central storage", $tags="logicApp")
Component(excelWatcher, "Excel Change Watcher", "Azure Logic App", "Monitors Excel changes and updates config data", $tags="logicApp")
   Component(blobExcel, "Excel File Store", "Azure Blob Storage", "Stores regional translation/configuration file", $tags="datastore") Component(tableStorage, "All Data Table", "Azure Table Storage", "Central store for all error cases and metadata", $tags="datastore")
   Component(sfPush, "Salesforce Connector", "Azure Logic App", "Asynchronously forwards structured data to Salesforce", $tags="logicApp")
Component(queryFunction, "Token Query Function", "Azure Function", "Retrieves error data for display on AEM page", $tags="azureFunction")
System_Ext(aem, "Adobe Experience Management Website", "Embedded React App requests token-based error info") System_Ext(sf, "Customer Relation System - Salesforce", "Receives error case data asynchronously")
System_Ext(excelUploader, "Excel Automation", "Uploads and manages regional Excel file")
Rel(tokenGenerator, urlConstructor, "Sends Token ID", "internal")
Rel(urlConstructor, dataCollector, "Sends Country URL + error data", "internal")
Rel(dataCollector, sfPush, "Prepares structured record", "internal")
Rel(dataCollector, tableStorage, "Stores token, URL, metadata", "Azur Rel(dataCollector, blobExcel, "Reads regional mappings", "Azure SDK")
Rel(excelWatcher, blobExcel, "Watches for changes", "Event Grid Trigger") Rel(excelWatcher, tableStorage, "Updates entries", "Azure SDK")
Rel(sfPush, sf, "Pushes support case", "HTTPS/REST - async")
' External Interfaces
```

```
Rel(queryFunction, tableStorage, "Queries data", "Azure SDK")
Rel(aem, queryFunction, "Requests token info", "HTTPS/JSON")

Rel(excelUploader, blobExcel, "Uploads Excel file", "HTTPS/Azure UI")
@enduml
```

This diagram cleanly shows:

- Each Logic App as a distinct component
- · Clear data flow between error processing, token generation, URL construction, and persistence
- Blob storage for Excel files, with synchronization
- Azure Function for AEM integration
- · Asynchronous push to Salesforce

Let me know if you'd like to continue with a code-level class or sequence diagram for any of these parts.

Du:

please make an Azure Enterprise Service Bus between everything that goes in and out outside of errro handling with the element AddElementTag("integrationBus", \$fontColor=\$ELEMENT\_FONT\_COLOR, \$bgColor="#335DA5", \$shape=EightSidedShape(), \$legendText="integration bus\n(eight sided)")

ChatGPT:

```
plantuml
   Kopieren
                             Bearbeiten
    ______
   C4 Component Diagram: Error Handling Module with Azure Service Bus
 !include https://raw.githubusercontent.com/plantuml-stdlib/C4-PlantUML/master/C4_Component.puml
    !include C4_Component.puml
LAYOUT WITH LEGEND()
title Error Handling Module - Component Diagram (with Azure Service Bus)
AddElementTag("logicApp", $fontColor=$ELEMENT_FONT_COLOR, $bgColor="#B0C4DE", $shape=RoundedBoxShape(), $legendText="Azure Logic App")

AddElementTag("azureFunction", $fontColor=$ELEMENT_FONT_COLOR, $bgColor="#DDA0DD", $shape=BoxShape(), $legendText="Azure Function")

AddElementTag("datastore", $fontColor=$ELEMENT_FONT_COLOR, $bgColor="#F5DEB3", $shape=EylinderShape(), $legendText="Data Store")

AddElementTag("integrationBus", $fontColor=$ELEMENT_FONT_COLOR, $bgColor="#335DA5", $shape=EightSidedShape(), $legendText="integration bus\n(eight sid
Container_Boundary(c1, "Error Handling Module") {
   Component(tokenGenerator, "Token Generator", "Azure Logic App", "Generates unique Token IDs", $tags="logicApp")
   Component(urlConstructor, "Country URL Builder", "Azure Logic App", "Creates country-specific URLs", $tags="logicApp")
   Component(dataCollector, "Data Collector", "Azure Logic App", "Routes error data to Salesforce and storage", $tags="logicApp")
   Component(excelWatcher, "Excel Change Watcher", "Azure Logic App", "Syncs Excel config changes", $tags="logicApp")
    Component(blobExcel, "Excel File Store", "Azure Blob Storage", "Holds Excel-based localization config", $tags="datastore") Component(tableStorage, "All Data Table", "Azure Table Storage", "Token-based error data store", $tags="datastore")
    Component(sfPush, "Salesforce Connector", "Azure Logic App", "Asynchronously sends data to Salesforce", $tags="logicApp")
Component(queryFunction, "Token Query Function", "Azure Function", "Serves token data to AEM", $tags="azureFunction")
System_Ext(aem, "Adobe Experience Management Website", "Embedded React app renders error info")
System_Ext(sf, "Customer Relation System - Salesforce", "Handles customer tickets")
System_Ext(excelUploader, "Excel Automation", "Uploads Excel config file")
Component(azureBus, "Azure Service Bus", "Enterprise Service Bus", "Integration channel", $tags="integrationBus")
Rel(tokenGenerator, urlConstructor, "Sends Token ID", "internal")
Rel(urlConstructor, dataCollector, "Sends enriched error record", "internal")
Rel(dataCollector, sfPush, "Forwards case structure", "internal")
Rel(dataCollector, tableStorage, "Stores token + metadata", "Azure SDK")
Rel(dataCollector, blobExcel, "Reads localization data", "Azure SDK")
Rel(excelWatcher, blobExcel, "Monitors Excel changes", "Event Grid") Rel(excelWatcher, tableStorage, "Updates data entries", "Azure SDK")
Rel(sfPush, azureBus, "Sends structured error info", "AMQP") Rel(azureBus, sf, "Delivers support case", "HTTPS/REST")
    External Interfaces
Rel(queryFunction, tableStorage, "Reads token data", "Azure SDK")
Rel(aem, azureBus, "Requests token info", "HTTPS/JSON")
Rel(azureBus, queryFunction, "Triggers data fetch", "HTTPS")
Rel(excelUploader, azureBus, "Uploads Excel file", "HTTPS") Rel(azureBus, blobExcel, "Stores uploaded file", "Blob API")
 ' Incoming from Order Processing
System_Ext(orderProcessing, "Order Processing Module", "Sends classified error data")
Rel(orderProcessing, azureBus, "Pushes error payload", "HTTPS/JSON")
Rel(azureBus, tokenGenerator, "Delivers error message", "AMQP")
@enduml
```

I want excel automation to be in error handling module and that it has relation to excel change watcher. Please do not make any elements except for the service bus. Plus please make the names nicer, and easy to understand

ChatGPT:

```
plantuml
   Kopieren
                               Bearbeiten
    C4 Component Diagram: Error Handling Module (with Azure Service Bus and Refined Naming)
 !include https://raw.githubusercontent.com/plantuml-stdlib/C4-PlantUML/master/C4 Component.puml
     !include C4_Component.puml
LAYOUT_WITH_LEGEND()
title Error Handling Module - Component Diagram (refined)
AddElementTag("logicApp", $fontColor=$ELEMENT_FONT_COLOR, $bgColor="#80C4DE", $shape=RoundedBoxShape(), $legendText="Azure Logic App")
AddElementTag("azureFunction", $fontColor=$ELEMENT_FONT_COLOR, $bgColor="#FDDADDD", $shape=BoxShape(), $legendText="Azure Function")

AddElementTag("datastore", $fontColor=$ELEMENT_FONT_COLOR, $bgColor="#F5DEB3", $shape=EylinderShape(), $legendText="Data Store")

AddElementTag("integrationBus", $fontColor=$ELEMENT_FONT_COLOR, $bgColor="#335DA5", $shape=EightSidedShape(), $legendText="integration bus\n(eight sides))
Container_Boundary(c1, "Error Handling Module") {
   Component(tokenLogicApp, "Token Generator", "Azure Logic App", "Generates a unique Token ID", $tags="logicApp")
   Component(urlLogicApp, "Country URL Builder", "Azure Logic App", "Combines Token ID with country-specific URL", $tags="logicApp")
   Component(dataCollector, "Data Collector", "Azure Logic App", "Forwards data to storage and Salesforce", $tags="logicApp")
   Component(excelWatcher, "Excel Sync Handler", "Azure Logic App", "Listens for changes to the Excel config", $tags="logicApp")
   Component(excelAutomation, "Excel Upload Automation", "Excel Script", "Maintains localization Excel file", $tags="logicApp")
    Component(blobExcel, "Excel File Storage", "Azure Blob Storage", "Holds uploaded Excel file", $tags="datastore")
Component(tableStorage, "Error Data Table", "Azure Table Storage", "Central storage for token-based error data", $tags="datastore")
    Component(sfForwarder, "Salesforce Forwarder", "Azure Logic App", "Sends structured data asynchronously", $tags="logicApp")
Component(queryFunction, "Token Query Function", "Azure Function", "Serves token-based data to frontend", $tags="azureFunction")
System_Ext(aem, "AEM Website (React)", "Displays error data and enables user input")
System_Ext(salesforce, "Salesforce", "Receives support cases")
Component(azureBus, "Azure Service Bus", "Enterprise Service Bus", "Message interface", $tags="integrationBus")
System_Ext(orderProcessing, "Order Processing Module", "Provides classified error payloads")
   Relations
Rel(orderProcessing, azureBus, "Pushes error data", "HTTPS")
Rel(azureBus, tokenLogicApp, "Delivers error record", "AMQP")
Rel(tokenLogicApp, urlLogicApp, "Sends Token ID")
Rel(urlLogicApp, dataCollector, "Sends Country URL and data")
Rel(dataCollector, sfForwarder, "Prepares payload")
Rel(dataCollector, tableStorage, "Stores tokenized record", "Azure SDK")
                                                               "Reads config", "Azure SDK")
Rel(dataCollector, blobExcel,
Rel(sfForwarder, azureBus, "Pushes case", "AMQP") Rel(azureBus, salesforce, "Delivers case", "HTTPS")
Rel(queryFunction, tableStorage, "Fetches data", "Azure SDK")
Rel(aem, azureBus, "Requests token info", "HTTPS")
Rel(azureBus, queryFunction, "Triggers query", "HTTPS")
Rel(excelAutomation, excelWatcher, "Sends file update")
Rel(excelWatcher, blobExcel, "Updates Excel data", "Azure SDK")
Rel(excelWatcher, tableStorage, "Syncs localized values", "Azure SDK")
@enduml
Du:
```

And use the containerdb for all databases and do not add any formatting except for integrationBus. all Logic apps send HTTP and JSON

ChatGPT

```
plantuml
    Kopieren
                                             Bearbeiten
      C4 Component Diagram: Error Handling Module (final formatting rules applied)
@startuml
!include https://raw.githubusercontent.com/plantuml-stdlib/C4-PlantUML/master/C4 Component.puml
       !include C4_Component.puml
LAYOUT_WITH_LEGEND()
title Error Handling Module - Component Diagram
Add Element Tag ("integration Bus", \$font Color=\$ELEMENT\_FONT\_COLOR, \$bg Color="\#335DA5", \$shape=Eight Sided Shape(), \$legend Text="integration bus \n(eight sided Shape(), \$legend Text="integration bus \n(eig
Container_Boundary(c1, "Error Handling Module") {
   Component(tokenLogicApp, "Token Generator", "Azure Logic App", "Generates a unique Token ID")
      Component(tokenLogicApp, "Token Generator", "Azure Logic App", "Generates a unique Token ID")

Component(urlLogicApp, "Country URL Builder", "Azure Logic App", "Combines Token ID with country-specific URL")

Component(dataCollector, "Data Collector", "Azure Logic App", "Forwards data to storage and Salesforce")

Component(excelWatcher, "Excel Sync Handler", "Azure Logic App", "Listens for changes to the Excel config")
      Component(excelAutomation, "Excel Upload Automation", "Excel Script", "Maintains localization Excel file")
      ContainerDb(blobExcel, "Excel File Storage", "Azure Blob Storage", "Holds uploaded Excel file")
ContainerDb(tableStorage, "Error Data Table", "Azure Table Storage", "Central storage for token-based error data")
       Component(sfForwarder, "Salesforce Forwarder", "Azure Logic App", "Sends structured data asynchronously")
Component(queryFunction, "Token Query Function", "Azure Function", "Serves token-based data to frontend")
```

```
System_Ext(aem, "AEM Website (React)", "Displays error data and enables user input")
System_Ext(salesforce, "Salesforce", "Receives support cases")
Component(azureBus, "Azure Service Bus", "Enterprise Service Bus", "Message interface", $tags="integrationBus")
System_Ext(orderProcessing, "Order Processing Module", "Provides classified error payloads")

' Relations
Rel(orderProcessing, azureBus, "Pushes error data", "HTTPS/JSON")
Rel(azureBus, tokenLogicApp, "Delivers error record", "AMQP")

Rel(tokenLogicApp, urlLogicApp, "Sends Token ID", "HTTPS/JSON")
Rel(urlLogicApp, dataCollector, "Sends enriched data", "HTTPS/JSON")
Rel(dataCollector, sfForwarder, "Prepares payload", "HTTPS/JSON")
Rel(dataCollector, tableStorage, "Stores tokenized record")
Rel(dataCollector, blobExcel, "Reads config")

Rel(sfForwarder, azureBus, "Pushes case", "HTTPS/JSON")
Rel(azureBus, salesforce, "Delivers case", "HTTPS/JSON")
Rel(azureBus, salesforce, "Delivers case", "HTTPS/JSON")
Rel(azureBus, queryFunction, tableStorage, "Fetches data")
Rel(acem, azureBus, "Requests token info", "HTTPS/JSON")
Rel(azureBus, queryFunction, "Triggers query", "HTTPS")

Rel(excelAutomation, excelWatcher, "Sends file update", "HTTPS/JSON")
Rel(excelAutomation, excelWatcher, "Sends file update", "HTTPS/JSON")
Rel(excelWatcher, blobExcel, "Updates Excel data")
Rel(excelWatcher, blobExcel, "Updates Excel data")
Rel(excelWatcher, tableStorage, "Syncs localized values")
```

is there a possibility to make it more optimized, i.e. all external containers next to each other graphically etc.?

ChatGPT:

Yes, optimizing the **layout of a PlantUML C4 diagram** for **graphical clarity**—like placing all external systems next to each other or enforcing specific topologies —is absolutely possible using PlantUML layout hints.

Here are techniques to improve the layout:

# Optimized Layout Techniques

#### 1. Use LAYOUT\_LEFT\_RIGHT()

This enforces a left-to-right orientation, which keeps input (e.g. external systems) on the left and output on the right—ideal for data flow systems.



#### 2. Use Together Blocks

This groups external systems or components visually adjacent:

#### 3. Use Direction Keywords (-down-, -left-, etc.) in Rel()

You can hint relative positioning using:

```
plantuml

Kopieren Bearbeiten

Rel_Down(aem, azureBus, "Requests token info", "HTTPS/JSON")

Rel_Right(azureBus, queryFunction, "Triggers query", "HTTPS")
```

# 4. Custom Layout via skinparam (Advanced)

For deeper control, you can use skinparam linetype ortho or other placement controls—but they're more involved and may override automatic layout rules in C4-PlantUML.

# *<del>colored Suggestion Suggestion* </del>

Would you like me to apply these optimizations directly to your component diagram? I can:

- Group all external systems together (left or top)
- Keep the service bus central
- Internal module below or right-aligned

Let me know and I'll generate a more layout-optimized version for you.

Du:

yes please

ChatGPT:

```
plantuml
                                            Bearbeiten
     Kopieren
       C4 Component Diagram: Error Handling Module (Optimized Layout)
 @startum1
 \=! include \ https://raw.githubusercontent.com/plantuml-stdlib/C4-PlantUML/master/C4\_Component.pumlercontent.com/plantuml-stdlib/C4-PlantUML/master/C4\_Component.pumlercontent.com/plantuml-stdlib/C4-PlantUML/master/C4\_Component.pumlercontent.com/plantuml-stdlib/C4-PlantUML/master/C4\_Component.pumlercontent.com/plantuml-stdlib/C4-PlantUML/master/C4\_Component.pumlercontent.com/plantuml-stdlib/C4-PlantUML/master/C4\_Component.pumlercontent.com/plantuml-stdlib/C4-PlantUML/master/C4\_Component.pumlercontent.com/plantuml-stdlib/C4-PlantUML/master/C4\_Component.pumlercontent.com/plantuml-stdlib/C4-PlantUML/master/C4\_Component.pumlercontent.com/plantuml-stdlib/C4-PlantUML/master/C4\_Component.pumlercontent.pumlercontent.pumlercontent.pumlercontent.pumlercontent.pumlercontent.pumlercontent.pumlercontent.pumlercontent.pumlercontent.pumlercontent.pumlercontent.pumlercontent.pumlercontent.pumlercontent.pumlercontent.pumlercontent.pumlercontent.pumlercontent.pumlercontent.pumlercontent.pumlercontent.pumlercontent.pumlercontent.pumlercontent.pumlercontent.pumlercontent.pumlercontent.pumlercontent.pumlercontent.pumlercontent.pumlercontent.pumlercontent.pumlercontent.pumlercontent.pumlercontent.pumlercontent.pumlercontent.pumlercontent.pumlercontent.pumlercontent.pumlercontent.pumlercontent.pumlercontent.pumlercontent.pumlercontent.pumlercontent.pumlercontent.pumlercontent.pumlercontent.pumlercontent.pumlercontent.pumlercontent.pumlercontent.pumlercontent.pumlercontent.pumlercontent.pumlercontent.pumlercontent.pumlercontent.pumlercontent.pumlercontent.pumlercontent.pumlercontent.pumlercontent.pumlercontent.pumlercontent.pumlercontent.pumlercontent.pumlercontent.pumlercontent.pumlercontent.pumlercontent.pumlercontent.pumlercontent.pumlercontent.pumlercontent.pumlercontent.pumlercontent.pumlercontent.pumlercontent.pumlercontent.pumlercontent.pumlercontent.pumlercontent.pumlercontent.pumlercontent.pumlercontent.pumlercontent.pumlercontent.pumlercontent.pumlercontent.pumlercontent.pumlercontent.pumlercontent.pumlercontent.pumlerconte
'!include C4_Component.puml
LAYOUT_LEFT_RIGHT()
title Error Handling Module - Component Diagram (optimized layout)
AddElementTag("integrationBus", $fontColor=$ELEMENT_FONT_COLOR, $bgColor="#335DA5", $shape=EightSidedShape(), $legendText="integration bus\n(eight sid
     External systems grouped
 together {
      System_Ext(orderProcessing, "Order Processing Module", "Provides classified error payloads")
System_Ext(aem, "AEM Website (React)", "Displays error data and enables user input")
System_Ext(salesforce, "Salesforce", "Receives support cases")
Component(azureBus, "Azure Service Bus", "Enterprise Service Bus", "Message interface", $tags="integrationBus")
     Internal boundary
Container_Boundary(c1, "Error Handling Module") {
   Component(tokenLogicApp, "Token Generator", "Azure Logic App", "Generates a unique Token ID")
   Component(urllogicApp, "Country URL Builder", "Azure Logic App", "Combines Token ID with country-specific U
   Component(dataCollector, "Data Collector", "Azure Logic App", "Forwards data to storage and Salesforce")
   Component(excelWatcher, "Excel Sync Handler", "Azure Logic App", "Listens for changes to the Excel config")
   Component(excelAutomation, "Excel Upload Automation", "Excel Script", "Maintains localization Excel file")
                                                                                                                                                                                                             "Combines Token ID with country-specific URL")
      ContainerDb(blobExcel, "Excel File Storage", "Azure Blob Storage", "Holds uploaded Excel file")
ContainerDb(tableStorage, "Error Data Table", "Azure Table Storage", "Central storage for token-based error data")
       Component(sfForwarder, "Salesforce Forwarder", "Azure Logic App", "Sends structured data asynchronously")
 Component(queryFunction, "Token Query Function", "Azure Function", "Serves token-based data to frontend")
Rel(orderProcessing, azureBus, "Pushes error data", "HTTPS/JSON")
Rel(azureBus, tokenLogicApp, "Delivers error record", "AMQP")
Rel(tokenLogicApp, urlLogicApp, "Sends Token ID", "HTTPS/JSON")
Rel(urlLogicApp, dataCollector, "Sends enriched data", "HTTPS/JSON")
Rel(dataCollector, sfForwarder, "Prepares payload", "HTTPS/JSON")
Rel(dataCollector, tableStorage, "Stores tokenized record")
Rel(dataCollector, blobExcel, "Reads config")
Rel(sfForwarder, azureBus, "Pushes case", "HTTPS/JSON") Rel(azureBus, salesforce, "Delivers case", "HTTPS")
Rel(queryFunction, tableStorage, "Fetches data")
Rel(aem, azureBus, "Requests token info", "HTTPS/JSON")
Rel(azureBus, queryFunction, "Triggers query", "HTTPS")
Rel(excelAutomation, excelWatcher, "Sends file update", "HTTPS/JSON")
Rel(excelWatcher, blobExcel, "Updates Excel data")
Rel(excelWatcher, tableStorage, "Syncs localized values")
@enduml
Du:
```

can you please also remember that if something sends a repsonse must also come in the relation

ChatGPT:

Absolutely — you're right. For systems involving request/response interactions (like AEM querying for token data), both directions must be explicitly modeled.

 $Here's \ the \ \textbf{corrected and response-complete version} \ of \ your \ diagram. \ The \ following \ request/response \ flows \ are \ now \ \textit{bidirectional} \ where \ applicable:$ 

- · Salesforce notifications assumed fire-and-forget, so left as-is
- Excel Automation sends only one-way update → Excel Sync Handler

```
LAYOUT LEFT RIGHT()
 title Error Handling Module - Component Diagram (with Responses)
AddElementTag("integrationBus", $fontColor=$ELEMENT FONT COLOR, $bgColor="#335DA5", $shape=EightSidedShape(), $legendText="integration bus\n(eight sid
 ' External systems grouped
 together {
    System_Ext(orderProcessing, "Order Processing Module", "Provides classified error payloads")
System_Ext(aem, "AEM Website (React)", "Displays error data and enables user input")
System_Ext(salesforce, "Salesforce", "Receives support cases")
Component(azureBus, "Azure Service Bus", "Enterprise Service Bus", "Message interface", $tags="integrationBus")
   Internal boundary
'Internal boundary
Container_Boundary(c1, "Error Handling Module") {
Component(tokenLogicApp, "Token Generator", "Azure Logic App", "Generates a unique Token ID")
Component(urllogicApp, "Country URL Builder", "Azure Logic App", "Combines Token ID with country-specific URL")
Component(dataCollector, "Data Collector", "Azure Logic App", "Forwards data to storage and Salesforce")
Component(excelWatcher, "Excel Sync Handler", "Azure Logic App", "Listens for changes to the Excel config")
Component(excelAutomation, "Excel Upload Automation", "Excel Script", "Maintains localization Excel file")
    ContainerDb(blobExcel, "Excel File Storage", "Azure Blob Storage", "Holds uploaded Excel file")
ContainerDb(tableStorage, "Error Data Table", "Azure Table Storage", "Central storage for token-based error data")
     Component(sfForwarder, "Salesforce Forwarder", "Azure Logic App", "Sends structured data asynchronously")
Component(queryFunction, "Token Query Function", "Azure Function", "Serves token-based data to frontend")
' Request/response and one-way flows
Rel(orderProcessing, azureBus, "Pushes error data", "HTTPS/JSON")
Rel(azureBus, tokenLogicApp, "Delivers error record", "AMQP")
Rel(tokenLogicApp, urlLogicApp, "Sends Token ID", "HTTPS/JSON")
Rel(urlLogicApp, dataCollector, "Sends enriched data", "HTTPS/JSON")
Rel(dataCollector, sfForwarder, "Prepares payload", "HTTPS/JSON")
Rel(dataCollector, tableStorage, "Stores tokenized record")
Rel(dataCollector, blobExcel, "Reads config")
Rel(sfForwarder, azureBus, "Pushes case", "HTTPS/JSON")
Rel(azureBus, salesforce, "Delivers case", "HTTPS")
 Rel(aem, azureBus, "Requests token info", "HTTPS/JSON")
Rel(azureBus, queryFunction, "Invokes query", "HTTPS")
Rel(queryFunction, azureBus, "Returns error data", "HTTPS/JSON")
Rel(azureBus, aem, "Delivers response", "HTTPS/JSON")
Rel(queryFunction, tableStorage, "Fetches data")
 Rel(excelAutomation, excelWatcher, "Sends file update", "HTTPS/JSON")
Rel(excelWatcher, blobExcel, "Updates Excel data")
Rel(excelWatcher, tableStorage, "Syncs localized values")
Let me know if you'd like to layer a sequence diagram for any part of this (e.g., AEM token flow, Excel sync, or Salesforce push).
```

Du

ChatGPT:

title order Processing Platform - Container Diagram (with Azure Service bus and Visual grouping)

```
Person(customer, "Customer", "Submits orders and receives error feedback")
  Group external systems horizontally
together {
   System_Ext(externalWebsite, "External Website", "Customer order entry point; performs basic file validation")
ContainerDb_Ext(externalDB, "External Database", "", "Stores order metadata temporarily after submission", $tags="db")
System_Boundary(s1, "Order Processing Platform") {
  ' Place Azure Bus in the center between external and internal
   Container(azureBus, "Azure Service Bus", "Azure Service Bus", "Secure message interface between external site and internal modules", $tags="integrat
      Group core processing on the left
      Container(orderProcessing, "Order Processing Module", "Microsoft Azure", "Analyzes uploaded documents using AI to detect errors")
ContainerDb(orderDB, "Order DB", "Azure Storage", "Stores processed order and document data", $tags="db")
      Group error handling on the right
   together {
      Container(errorHandling, "Error Handling Module", "Azure Logic Apps & Azure Functions", "Applies validation rules and manages error forwarding")
ContainerDb(errorDB, "Error DB", "Azure Storage", "Stores validation outcomes and tracking tokens", $tags="db")
      Place interfaces centrally
   together {
      vactor (
Container(salesforce, "Customer Relation System - Salesforce", "Salesforce", "Manages internal tickets and communicates with customers")
Container(aemInterface, "Adobe Experience Management Website", "Adobe Experience Manager", "Provides error visibility and correction submission vi
Rel(customer, externalWebsite, "Submits order with document", "HTTPS")
Rel(externalWebsite, externalDB, "Stores order metadata", "Azure SDK/HTTPS")
Rel(externalWebsite, azureBus, "Sends document for processing", "HTTPS/JSON")
Rel(azureBus, orderProcessing, "Forwards document", "AMQP/HTTPS")
Rel(orderProcessing, orderDB, "Reads/writes order and document data", "Azure SDK"
Rel(orderProcessing, errorHandling, "Passes classified error data", "HTTPS/JSON")
Rel(errorHandling, errorDB, "Stores validation errors and token IDs", "Azure SDK")
Rel(errorHandling, salesforce, "Sends validation results and tracking token", "HTTPS/REST")
Rel(aemInterface, errorHandling, "Reads error data for token-based display", "HTTPS/RES Rel(aemInterface, salesforce, "Submits corrections and support requests", "HTTPS/REST") Rel(customer, aemInterface, "Views errors and submits corrections using token", "HTTPS"
Rel(customer, aemInterface, "Views errors and submits corrections using token", "HTTPS") Rel(salesforce, customer, "Sends ticket updates and status notifications", "Email/HTTPS")
@endum1
Du:
                                                                        = 'C4 Context Diagram: Order Processing Platform (with Protocols)' =
that too, please '=
@startuml !include https://raw.githubusercontent.com/plantuml-stdlib/C4-PlantUML/master/C4 Component.puml '!include C4 Component.puml
LAYOUT_WITH_LEGEND() title Order Processing Platform - Context Diagram Person(customer", "Customer", "Submits orders and receives error feedback")
System_Ext(externalWebsite, "External Website", "Customer order entry point; performs basic file validation") System_Boundary(s1, "Order Processing Platform")
{ Container(orderProcessing, "Order Processing Module", "Microsoft Azure", "Analyzes uploaded documents using AI to detect errors") Container(errorHandling,
"Error Handling Module", "Azure Logic Apps & Azure Functions", "Applies validation rules and manages error forwarding") Container(salesforce, "Customer Relation System - Salesforce", "Manages internal tickets and communicates with customers") Container(aemInterface, "Adobe Experience
Management Website", "Adobe Experience Manager", "Provides error visibility and correction submission via token") Rel(customer, external Website, "Submits order with document", "HTTPS") Rel(external Website, order Processing, "Transfers validated document", "HTTPS") Rel(order Processing, error Handling, "Passes
classified error data if issues detected", "HTTPS/JSON") Rel(errorHandling, salesforce, "Sends validation results and tracking token", "HTTPS/REST") Rel(aemInterface, errorHandling, "Reads error data for token-based display", "HTTPS/REST") Rel(aemInterface, salesforce, "Submits corrections and support
requests", "HTTPS/REST") Rel(customer, aemInterface, "Views errors and submits corrections using token", "HTTPS") Rel(salesforce, customer, "Sends ticket
updates and status notifications", "Email/HTTPS") @enduml
ChatGPT:
plantuml
                       Bearbeiten
  Kopieren
   C4 Context Diagram: Order Processing Platform (Optimized Visual Layout with Protocols)
@startuml
 .
!include https://raw.githubusercontent.com/plantuml-stdlib/C4-PlantUML/master/C4_Component.puml
'!include C4_Component.puml
LAYOUT_LEFT_RIGHT()
title Order Processing Platform - Context Diagram (optimized layout)
 ' Group external systems horizontally
together {
   Person(customer, "Customer", "Submits orders and receives error feedback")
System_Ext(externalWebsite, "External Website", "Customer order entry point; performs basic file validation")
System_Boundary(s1, "Order Processing Platform") {
   ' Group internal containers logically together {
       Gether (Container(orderProcessing, "Order Processing Module", "Microsoft Azure", "Analyzes uploaded documents using Ai to detect entry (Container(errorHandling, "Error Handling Module", "Azure Logic Apps & Azure Functions", "Applies validation rules and manages error forwarding")

Container(errorHandling, "Error Handling Module", "Azure Logic Apps & Azure Functions", "Applies validation rules and manages error forwarding")
      Container (errorHandling,
   together {
      Container(salesforce, "Customer Relation System - Salesforce", "Salesforce", "Manages internal tickets and communicates with customers")
Container(aemInterface, "Adobe Experience Management Website", "Adobe Experience Manager", "Provides error visibility and correction submission vi
```

```
Relationships
Rel(customer, externalWebsite, "Submits order with document", "HTTPS")
Rel(externalWebsite, orderProcessing, "Transfers validated document", "HTTPS")

Rel(orderProcessing, errorHandling, "Passes classified error data if issues detected", "HTTPS/JSON",
Rel(errorHandling, salesforce, "Sends validation results and tracking token", "HTTPS/REST")

Rel(aemInterface, errorHandling, "Reads error data for token-based display", "HTTPS/REST")
Rel(aemInterface, salesforce, "Submits corrections and support requests", "HTTPS/REST")
Rel(customer, aemInterface, "Views errors and submits corrections using token", "HTTPS")
Rel(salesforce, customer, "Sends ticket updates and status notifications", "Email/HTTPS")

@enduml
```

so that Customer is at the top center and below it external website and in the box directly above order processing, below error handling

ChatGPT:



okay, now give me code level with all the information you have right now, i need attributes and operations, so uml class diagram '

"include https://raw.githubusercontent.com/plantuml-stdlib/C4-PlantUML/master/C4\_Component.puml !'.include C4\_Component.puml LAYOUT\_LEFT\_RIGHT()
title Error Handling Module - Component Diagram (with Responses) AddElementTag("integrationBus", \$fontColor=\$ELEMENT\_FONT\_COLOR,
\$bgColor="#335DA5", \$shape=EightSidedShape(), \$legendText="integration bus\n(eight sided)") 'External systems grouped together {
System\_Ext(orderProcessing, "Order Processing Module", "Provides classified error payloads") \$System\_Ext(aem, "AEM Website (React)", "Displays error data and enables user input") \$System\_Ext(salesforce, "Salesforce", "Receives support cases") } Component(azureBus, "Azure Service Bus", "Enterprise Service Bus",
"Message interface", \$tags="integrationBus") 'Internal boundary Container\_Boundary(c1, "Error Handling Module") { Component(tokenLogicApp, "Token
Generator", "Azure Logic App", "Generates a unique Token ID") Component(urlLogicApp, "Country URL Builder", "Azure Logic App", "Combines Token ID with
country-specific URL") Component(dataCollector, "Data Collector", "Azure Logic App", "Forwards data to storage and Salesforce") Component(excelWatcher,
"Excel Sync Handler", "Azure Logic App", "Listens for changes to the Excel config") Component(excelAutomation, "Excel Upload Automation", "Excel Script",
"Maintains localization Excel file") ContainerDb(blobExcel, "Excel File Storage", "Azure Blob Storage", "Holds uploaded Excel file") ContainerDb(tableStorage,
"Error Data Table", "Azure Table Storage", "Central storage for token-based error data") Component(sfforwarder, "Salesforce Forwarder", "Azure Logic App",
"Request/response and one-way flows Rel(orderProcessing, azureBus, "Pushes error data", "HTTPS/JSON") Rel(azureBus, token-based data to frontend") '
Request/response and one-way flows Rel(orderProcessing, azureBus, "Pushes error data", "HTTPS/JSON") Rel(azureBus, solenticed data", "HTTPS/JSON")
Rel(dataCollector, sfForwarder, "Prepares payload", "HTTPS/JSON") Rel(dataCollector, tableStorage, "Stores tokenized r

ChatGPT:

Here is the code-level UML class diagram (@startuml to @enduml) representing the Error Handling Module.

It includes:

- All relevant classes for Logic Apps, the Azure Function, and storage entities
- Attributes based on the known data model (e.g., token, metadata, URLs)

```
plantuml
 O<sub>Kopieren</sub>
                Bearbeiten
 UML Class Diagram: Error Handling Module - Code Level
@startuml
title Error Handling Module - Code-Level UML Diagram
class TokenGenerator {
  +String documentId
+String timestamp
  +String generateToken()
  +boolean validatePayload()
class CountryUrlBuilder {
  +String token
  +String countryCode
  +String baseUrl
  +String buildUrl()
class DataCollector {
  +String token
+String countryUrl
  +List<String> errorCodes
+Map<String,String> metadata
  +void enrichWithExcelData()
  +void writeToTableStorage()
+void forwardToSalesforce()
class ExcelSyncHandler {
  +String blobName
+DateTime lastModified
  +void watchForChanges()
  +void syncToTableStorage()
class ExcelUploadAutomation {
  +String localPath
  +String uploadTimestamp
  +void uploadToBlob()
class SalesforceForwarder {
  +String caseId
  +Map<String,String> payload
  +void sendCaseData()
class TokenQueryFunction {
  +String token
  +Map<String,String> queryByToken()
class BlobStorage {
  +String fileName
  +String container
  +void readFile()
  +void writeFile()
class TableStorage {
  +String partitionKey
  +String rowKey
  +String token
  +String countryUrl
  +List<String> errorCodes
+Map<String,String> metadata
  +void insertRecord()
  +Map<String,String> getRecordByToken(String token)
class AzureServiceBus {
  +void receiveMessage()
  +void sendMessage()
class OrderProcessingModule {
  +void sendErrorData()
class AEMFrontend {
  +void requestErrorData(String token)
  +void displayData(Map<String,String>)
```

```
class Salesforce {
  +void receiveCase()
  +void notifyCustomer()
 Relationships
OrderProcessingModule --> AzureServiceBus : sendErrorData()
AzureServiceBus --> TokenGenerator : deliverMessage()
TokenGenerator --> CountryUrlBuilder : generateToken()
CountryUrlBuilder --> DataCollector : buildUrl()
DataCollector --> TableStorage : writeToTableStorage()
DataCollector --> SalesforceForwarder : forwardToSalesforce()
DataCollector --> BlobStorage : enrichWithExcelData()
ExcelUploadAutomation --> ExcelSvncHandler : uploadToBlob()
ExcelSyncHandler --> BlobStorage : watchForChanges()
ExcelSyncHandler --> TableStorage : syncToTableStorage()
SalesforceForwarder --> AzureServiceBus : sendCaseData()
AzureServiceBus --> Salesforce : receiveCase()
AEMFrontend --> AzureServiceBus : requestErrorData()
AzureServiceBus --> TokenQueryFunction : invokeQuery()
TokenQueryFunction --> AzureServiceBus : returnData()
TokenQueryFunction --> TableStorage : queryByToken()
AzureServiceBus --> AEMFrontend : deliverData()
Salesforce --> Customer : notifyCustomer()
@endum1
```

Let me know if you'd like to generate a **sequence diagram** (e.g., for a specific use case like token resolution or Excel sync) or need this split per component class file design.

#### Du:

can you please reframe errorhanling classes so that one understands what is outside and what is inside

ChatGPT

Certainly. To clearly distinguish between:

- Internal system components (within the Error Handling Module)
- External systems (like Salesforce, AEM, etc.)
- Infrastructure layers (like Azure Service Bus, Blob, Table)

I'll apply these principles:

# **☑** Naming and Visual Conventions

- Prefix internal classes with EH\_ (for Error Handling)
- Group external classes with a comment header: // External Systems
- Group infrastructure classes with: // Infrastructure
- · Keep class responsibilities and method names intuitive

### Refined UML Class Diagram

```
plantuml
 To Kopieren | Do Bearbeiten
 UML Class Diagram: Error Handling Module (refined structure)
@startuml
title Error Handling Module - Code-Level UML (clearly scoped)
  === Internal Components (EH_*)
class EH_TokenGenerator {
  +String documentId
  +String timestamp
  +String generateToken()
  +boolean validatePayload()
class EH_CountryUrlBuilder {
  +String token
  +String countryCode
  +String baseUrl
 +String buildUrl()
class EH DataCollector {
  +String token
  +String countryUrl
  +List<String> errorCodes
```

```
+Map<String,String> metadata
  +void enrichWithExcelData()
  +void writeToTable()
  +void sendToSalesforce()
class EH ExcelSyncHandler {
  +String blobName
  +DateTime lastModified
  +void watchChanges()
  +void updateTable()
class EH_ExcelUploader {
    +String localFilePath
  +void uploadExcelFile()
class EH_SalesforceForwarder \{
  +String caseId
+Map<String,String> payload
  +void forwardCase()
class EH_TokenQueryFunction {
  +String token
  +Map<String,String> getErrorDetails()
' === Infrastructure
class AzureServiceBus {
  +void sendMessage()
  +String receiveMessage()
class AzureBlobStorage {
  +String fileName
  +String read()
  +void write()
class AzureTableStorage {
  +String partitionKey
  +String rowKey
  +Map<String,String> errorRecord
  +void insert()
  +Map<String,String> query(String token)
' === External Systems
class OrderProcessingModule {
  +void sendErrorData()
class AEMFrontend {
  +void requestErrorData(String token)
  +void display(Map<String,String>)
class Salesforce {
  +void createCase()
  +void sendNotification()
class Customer {
  +void submitCorrection()
  +void receiveNotification()
' === Relations
OrderProcessingModule --> AzureServiceBus : sendErrorData()
AzureServiceBus --> EH_TokenGenerator : receiveMessage()
EH_TokenGenerator --> EH_CountryUrlBuilder : generateToken()
EH_CountryUrlBuilder --> EH_DataCollector : buildUrl()
EH_DataCollector --> AzureTableStorage : writeToTable()
EH_DataCollector --> EH_SalesforceForwarder : sendToSalesforce()
EH_DataCollector --> AzureBlobStorage : enrichWithExcelData()
EH_ExcelUploader --> EH_ExcelSyncHandler : uploadExcelFile()
EH_ExcelSyncHandler --> AzureBlobStorage : watchChanges()
EH_ExcelSyncHandler --> AzureTableStorage : updateTable()
EH_SalesforceForwarder --> AzureServiceBus : forwardCase()
AzureServiceBus --> Salesforce : createCase()
AEMFrontend --> AzureServiceBus : requestErrorData()
AzureServiceBus --> EH_TokenQueryFunction : getErrorDetails()
EH_TokenQueryFunction --> AzureTableStorage : query()
```

```
EH_TokenQueryFunction --> AzureServiceBus : returnData()
AzureServiceBus --> AEMFrontend : deliverData()
Salesforce --> Customer : sendNotification()
Customer --> AEMFrontend : submitCorrection()
@endum1
```

This version explicitly:

- Tags internal components with EH\_ to show they're part of Error Handling
- Groups all Azure services under Infrastructure
- Groups external systems separately
- Makes data flow and direction readable and modular

Let me know if you'd like a sequence diagram for any of these flows.

#### Du:

name it much more by class instead of technology and write the technology behind it in brackets or like you do in c4 modeling. Plus add whether it is public or private etc. and which calls have to be made with http etc. please really go into detail about the classes, so make it detailed

ChatGPT:

Excellent — let's now refactor and enrich the UML class diagram to align with your request. This version will:

# **✓** Structure

- 1. Use class names based on domain responsibilities, not implementation tech.
- 2. Indicate the underlying technology in parentheses, like in C4 (Logic App, Azure Function, Blob Storage, etc.).
- 3. Specify access level:
  - o + public
  - # protected
  - o private
- 4. Annotate operations with HTTP (or protocol) where relevant.
- 5. Be explicit about internal vs. external vs. infrastructure.

# Refined and Detailed Code-Level UML Class Diagram

```
plantuml
              **Bearbeiten
 Kopieren
 UML Class Diagram: Error Handling Module (Fully Detailed & C4-Styled)
@startuml
title Error Handling Module - Detailed Code-Level UML
 INTERNAL DOMAIN CLASSES (EH_)
class TokenService <<Azure Logic App>> {
   String documentId
  - Date createdAt
   String generateToken()
                                       <<internal>>
    boolean validateErrorPayload()
                                       <<internal>>
class UrlBuilderService <<Azure Logic App>> {
    String token
    String countryCode
   String baseUrl
   String buildUrl()
                                       <<internal>>
class ErrorDataCollector <<Azure Logic App>> {
   String token
    String countryUrl
    List<String> errorCodes
  - Map<String, String> metadata
   void enrichWithLocalizationData() <<reads from Blob Storage>>
  + void storeErrorRecord()
                                       <<writes to Table Storage>>
    void prepareCasePayload()
                                       <<internal>>
    void forwardToSalesforce()
class LocalizationSyncService <<Azure Logic App>> {
  - String fileName
```

```
+ void monitorExcelUpdates()
                                          <<triggered by Event Grid>>
                                      <<pre><<triggered by L.L...
<<writes to Table Storage>>
  + void syncDataToTableStorage()
class ExcelAutomationScript <<Excel Script (private)>> {
  - String localPath
                                      <<HTTPS PUT to Blob>>
  + void uploadLocalizationFile()
class SalesforceIntegrationService <<Azure Logic App>> {
  - String caseId
  - Map<String,String> structuredPayload
  + void pushToSalesforce()
                                          <<HTTP POST>>
class TokenQueryApi <<Azure Function (public)>> {
  + Map<String,String> getErrorDataByToken(String token) <<HTTP GET>>
  INFRASTRUCTURE SERVICES
class AzureServiceBus <<Enterprise Service Bus (internal)>> {
  + void sendMessage(Object payload)
  + Object receiveMessage()
class ErrorDataStore <<Azure Table Storage (internal)>> {
  - String partitionKey
  - String rowKey
  - Map<String, String> attributes
  + void insertRecord()
  + Map<String, String> fetchRecordByToken(String token)
class LocalizationBlobStore <<Azure Blob Storage (internal)>> {
  - String blobFile
  + String readExcelFile()
+ void writeExcelFile()
 EXTERNAL SYSTEMS
class OrderAnalysisEngine <<Order Processing (external)>> {
                                          <<HTTPS POST to Service Bus>>
  + void sendErrorPayload()
class AEMFrontendApp <<React SPA (external)>> {
  + void requestTokenData(String token) <<HTTPS GET>>
    void renderErrorDetails()
  + void redirectToCorrectionForm()
class SalesforceCRM <<CRM (external)>> {
  + void receiveCase()
  + void sendCustomerNotification() <<Email/REST>>
class Customer <<End User (external)>> {
  + void uploadDocument()
    void viewErrorDetails()
    void submitCorrection()
' RELATIONSHIPS (DETAILED)
OrderAnalysisEngine --> AzureServiceBus : sendErrorPayload() <<HTTPS>> AzureServiceBus --> TokenService : receiveMessage()
TokenService --> UrlBuilderService : generateToken(
UrlBuilderService --> ErrorDataCollector : buildUrl()
ErrorDataCollector --> LocalizationBlobStore : enrichWithLocalizationData()
ErrorDataCollector --> ErrorDataStore : storeErrorRecord()
ErrorDataCollector --> SalesforceIntegrationService : prepareCasePayload()
SalesforceIntegrationService --> AzureServiceBus : pushToSalesforce() <<HTTP>>>
AzureServiceBus --> SalesforceCRM : receiveCase() <<HTTP>>
ExcelAutomationScript --> LocalizationSyncService : uploadLocalizationFile() <<HTTPS>>
LocalizationSyncService --> LocalizationBlobStore : monitorExcelUpdates() LocalizationSyncService --> ErrorDataStore : syncDataToTableStorage()
```

- DateTime lastModified

```
AEMFrontendApp --> AzureServiceBus : requestTokenData() <<HTTPS>> AzureServiceBus --> TokenQueryApi : getErrorDataByToken() <<HTTP>> TokenQueryApi --> ErrorDataStore : fetchRecordByToken() TokenQueryApi --> AzureServiceBus : returnResponse() AzureServiceBus --> AEMFrontendApp : deliverTokenData()

SalesforceCRM --> Customer : sendCustomerNotification() Customer --> AEMFrontendApp : submitCorrection()

@enduml
```

### **What This Provides**

- Clear domains: internal vs. infrastructure vs. external
- Explicit protocol use: HTTPS, Event Grid, Blob API, etc.
- · Responsibility-based class naming
- Class-level technology context (C4-style annotations like <<Azure Logic App>>)
- · Correct visibility markers for attributes and methods

Would you like to split this into **physical modules**, **deployment diagram**, or continue with a **sequence diagram** for one of the workflows (e.g., "Token Lookup from AEM")?

Du:



can you please include other parameters that you think make sense for the classes? To make it more detailed

ChatGPT:

Great! I've reviewed the UML class diagram image you provided. It already includes detailed domain models like:

- Chat
- ErrorClassification
- ErrorInformation
- Token
- SalesforceCase
- EvaluationLog
- CountryLocalization, etc.

To enhance detail and practical value for each class, I recommend extending the following dimensions:

# **✓** Additional Parameters to Include (Domain-Wise)

#### General Entity Metadata

• createdAt: DateTime

• updatedAt: DateTime

• createdBy: String (e.g., system name or user ID)

• sourceSystem: String (if multiple sources feed data)

### Error Domain Enhancements

 $For \ {\tt ErrorInformation} \ / \ {\tt ErrorClassification} :$ 

- severity: String (e.g., INFO, WARNING, CRITICAL)
- ruleId: String (reference to validation rule)
- resolutionStatus: Enum { unresolved, pending, resolved }
- errorCategory: Enum (e.g., Format, Logic, MissingData)

# Token & Access Tracking

- tokenStatus: Enum { active, expired, invalid }
- lastAccessed: DateTime
- accessCount: Integer
- accessedByIP: String

#### SalesforceCase / CRM Integration

- sfCaseId: String (external reference)
- syncStatus: Enum { pending, synced, failed }
- lastSyncedAt: DateTime
- assignedAgent: String
- priority: Enum { low, medium, high }

## AEM + Frontend Usage Tracking

- viewedBy: List<User>
- submittedCorrections: Integer
- formRedirectUrl: String
- feedbackProvided: Boolean

# EvaluationLog Enhancements

- evaluationOutcome: Enum { success, fail, warning }
- evaluationDurationMs: Long
- executedRuleSetVersion: String
- triggerSource: Enum { auto, manual, scheduled }

## • CountryLocalization

- countryName: String
- supportedLanguages: List<String>
- defaultTimezone: String

# Suggested Attributes per Class Example

#### ErrorInformation



- + errorCode: String + errorMessage: String + severity: String
- + resolutionStatus: Enum
- + ruleId: String
  + category: String
- + createdAt: DateTime
- Token

# plaintext Kopieren Bearbeiten

- + tokenId: String + tokenStatus: Enum
- + issuedAt: DateTime
- + expiresAt: DateTime
  + lastAccessed: DateTime

- + accessCount: Integer + associatedChatId: UUID

#### SalesforceCase

# plaintext Nopieren Bearbeiten + sfCaseId: String + internalReferenceId: UUID

- + status: Enum
- + syncStatus: Enum

- + assignedAgent: String + priority: Enum + lastSyncedAt: DateTime

#### **EvaluationLog**

plaintext \*\*Bearbeiten Kopieren

```
+ logId: UUID
+ evaluationOutcome: Enum
+ ruleSetVersion: String
+ startedAt: DateTime
+ durationMs: Long
+ triggerSource: Enum
```

Would you like me to deliver a revised UML PlantUML code that includes all these enriched class definitions and relationships?

#### Du:

```
please add it to the code level '===
                                                      ====== ' UML Class Diagram: Error Handling Module (Fully Detailed & C4-Styled) '
                               = @startuml title Error Handling Module – Detailed Code-Level UML' =
                                                                                                                                  ='INTERNAL
                                   ========class TokenService <<Azure Logic App>> { - String documentId - Date createdAt -- + String
DOMAIN CLASSES (EH ) ' ===
generateToken() <<internal>> + boolean validateErrorPayload() <<internal>> } class UrlBuilderService <<Azure Logic App>> { - String token - String
countryCode - String baseUrl -- + String buildUrl() <<i internal >> } class ErrorDataCollector <<Azure Logic App>> { - String token - String countryUrl -
List<String> errorCodes - Map<String, String> metadata -- + void enrichWithLocalizationData() << reads from Blob Storage>> + void storeErrorRecord() << writes
to Table Storage>> + void prepareCasePayload() <<internal>> + void forwardToSalesforce() <<HTTP POST>> } class LocalizationSyncService <<Azure Logic
App>> { - String fileName - DateTime lastModified -- + void monitorExcelUpdates() << triggered by Event Grid>> + void syncDataToTableStorage() << writes to
Table Storage>> } class ExcelAutomationScript <<Excel Script (private)>> { - String localPath - Date lastUpload -- + void uploadLocalizationFile() <<HTTPS PUT
to Blob>> } class SalesforceIntegrationService <<Azure Logic App>> { - String caseId - Map<String> structuredPayload -- + void pushToSalesforce()
<<HTTP POST>> } class TokenQueryApi <<Azure Function (public)>> { -- + Map<String, String> getErrorDataByToken(String token) <<HTTP GET>> }
                                = ' INFRASTRUCTURE SERVICES ' =
                                                                                                    = class AzureServiceBus <<Enterprise Service Bus
(internal)>> { -- + void sendMessage(Object payload) + Object receiveMessage() } class ErrorDataStore << Azure Table Storage (internal)>> { - String partitionKey
- String rowKey - Map<String, String> attributes -- + void insertRecord() + Map<String> fetchRecordByToken(String token) } class LocalizationBlobStore
<<Azure Blob Storage (internal)>> { - String blobFile -- + String readExcelFile() + void writeExcelFile() } ' =
                                           = class OrderAnalysisEngine <<Order Processing (external)>> { -- + void sendErrorPayload() <<HTTPS POST to
SYSTEMS '=
Service Bus>> } class AEMFrontendApp <<React SPA (external)>> { -- + void requestTokenData(String token) <<HTTPS GET>> + void renderErrorDetails() +
void redirectToCorrectionForm() } class SalesforceCRM <<CRM (external)>> { -- + void receiveCase() <<HTTPS POST>> + void sendCustomerNotification()
<<Email/REST>> } class Customer <<End User (external)>> { -- + void uploadDocument() + void viewErrorDetails() + void submitCorrection() } '
                               = ' RELATIONSHIPS (DETAILED) ' =
                                                                                                 == OrderAnalysisEngine --> AzureServiceBus :
sendErrorPayload() << HTTPS>> AzureServiceBus --> TokenService : receiveMessage() TokenService --> UrlBuilderService : generateToken() UrlBuilderService --
> ErrorDataCollector : buildUrl() ErrorDataCollector --> LocalizationBlobStore : enrichWithLocalizationData() ErrorDataCollector --> ErrorDataStore :
storeErrorRecord() ErrorDataCollector --> SalesforceIntegrationService : prepareCasePayload() SalesforceIntegrationService --> AzureServiceBus :
pushToSalesforce() << HTTP>> AzureServiceBus --> SalesforceCRM : receiveCase() << HTTP>> ExcelAutomationScript --> LocalizationSyncService :
uploadLocalizationFile() << HTTPS>> LocalizationSyncService --> ErrorDataStore : monitorExcelUpdates() LocalizationSyncService --> ErrorDataStore :
syncDataToTableStorage() AEMFrontendApp --> AzureServiceBus : requestTokenData() << HTTPS>> AzureServiceBus --> TokenQueryApi :
getErrorDataByToken() <<HTTP>> TokenQueryApi --> ErrorDataStore : fetchRecordByToken() TokenQueryApi --> AzureServiceBus : returnResponse()
AzureServiceBus --> AEMFrontendApp : deliverTokenData() SalesforceCRM --> Customer : sendCustomerNotification() Customer --> AEMFrontendApp :
submitCorrection() @enduml
```

#### ChatGPT:

Here is your enhanced **code-level UML class diagram**, now extended with **detailed attributes** for core entities based on your previous diagram and best-practice suggestions.

```
plantuml
 Kopieren
              Bearbeiten
  -----
  UML Class Diagram: Error Handling Module (Fully Detailed + Enriched)
@startuml
title Error Handling Module - Code-Level UML (Detailed with Attributes & Metadata)
  INTERNAL DOMAIN CLASSES (EH )
class TokenService <<Azure Logic App>> {
  - String documentId
   Date createdAt
   String tokenStatus
                                      <<Enum { active, expired, invalid }>>
   String generateToken()
                                       <<internal>>
   boolean validateErrorPavload()
                                       <<internal>>
class UrlBuilderService <<Azure Logic App>> {
   String token
   String countryCode
    String baseUrl
    String countryUrl
  + String buildUrl()
                                       <<internal>>
class ErrorDataCollector <<Azure Logic App>> {
    String token
    String countryUrl
   List<String> errorCodes
Map<String, String> metadata
    String ruleId
    String severity
                                      <<Enum { info, warning, critical }>>
    String resolutionStatus
                                      <<Enum { unresolved, pending, resolved }>>
                                      <<Enum { format, logic, missingData }>>
   String errorCategory
    void enrichWithLocalizationData() <<reads from Blob Storage>>
    void storeErrorRecord()
                                       <<writes to Table Storage>>
    void prepareCasePavload()
                                       <<internal>>
        forwardToSalesforce()
                                       <<HTTP POST>>
```

```
class LocalizationSyncService <<Azure Logic App>> {
   String fileName
  - DateTime lastModified
  - String updatedBy
  + void monitorExcelUpdates()
                                       <<triggered by Event Grid>>
                                       <<writes to Table Storage>>
  + void syncDataToTableStorage()
class ExcelAutomationScript <<Excel Script (private)>> {
  - String localPath
  - Date lastUpload
 + void uploadLocalizationFile()
                                    <<HTTPS PUT to Blob>>
class SalesforceIntegrationService <<Azure Logic App>> {
  - String caseId
  - Map<String,String> structuredPayload
  - String syncStatus
                                      <<Enum { pending, synced, failed }>>
  - DateTime lastSyncedAt
 - String assignedAgent
  - String priority
                                     <<Enum { low, medium, high }>>
  + void pushToSalesforce()
                                      <<HTTP POST>>
class TokenQueryApi <<Azure Function (public)>> {
  - String requestIp
  - DateTime lastAccessed
  - Integer accessCount
  + Map<String,String> getErrorDataByToken(String token) <<HTTP GET>>
  _____
' INFRASTRUCTURE SERVICES
class AzureServiceBus <<Enterprise Service Bus (internal)>> {
  + void sendMessage(Object payload)
+ Object receiveMessage()
class ErrorDataStore <<Azure Table Storage (internal)>> {
  - String partitionKey
  - String rowKey
  - String token
  - String countryUrl
 List<String> errorCodesMap<String, String> metadata
 - DateTime createdAt
  - DateTime updatedAt
  + void insertRecord()
  + Map<String, String> fetchRecordByToken(String token)
class LocalizationBlobStore <<Azure Blob Storage (internal)>> {
 - String blobFile
   String uploaderId
  - String region
  + String readExcelFile()
  + void writeExcelFile()
' EXTERNAL SYSTEMS
class OrderAnalysisEngine <<Order Processing (external)>> {
  + void sendErrorPayload()
                                     <<hr/>HTTPS POST to Service Bus>>
class AEMFrontendApp <<React SPA (external)>> {
   String formRedirectUrl
   boolean feedbackProvided
  - Integer \operatorname{submittedCorrections}
 + void requestTokenData(String token) <<HTTPS GET>>
  + void renderErrorDetails()
   void redirectToCorrectionForm()
class SalesforceCRM <<CRM (external)>> {
 <<HTTPS POST>>
class Customer <<End User (external)>> {
 - String sessionId
- String email
 + void uploadDocument()
  + void viewErrorDetails()
  + void submitCorrection()
```

```
RELATIONSHIPS (DETAILED)
OrderAnalysisEngine --> AzureServiceBus : sendErrorPayload() <<HTTPS>>
AzureServiceBus --> TokenService : receiveMessage()
TokenService --> UrlBuilderService : generateToken()
UrlBuilderService --> ErrorDataCollector : buildUrl()
ErrorDataCollector --> LocalizationBlobStore : enrichWithLocalizationData()
ErrorDataCollector --> ErrorDataStore : storeErrorRecord()
ErrorDataCollector --> SalesforceIntegrationService : prepareCasePayload()
SalesforceIntegrationService --> AzureServiceBus : pushToSalesforce() <<HTTP>>>
AzureServiceBus --> SalesforceCRM : receiveCase() <<HTTP>>
ExcelAutomationScript --> LocalizationSyncService : uploadLocalizationFile() <<HTTPS>>
LocalizationSyncService --> LocalizationBlobStore : monitorExcelUpdates() LocalizationSyncService --> ErrorDataStore : syncDataToTableStorage()
AEMFrontendApp --> AzureServiceBus : requestTokenData() <<HTTPS>>
AzureServiceBus --> TokenQueryApi : getErrorDataByToken() <<HTTP>>
TokenQueryApi --> ErrorDataStore : fetchRecordByToken()
TokenQueryApi --> AzureServiceBus : returnResponse()
AzureServiceBus --> AEMFrontendApp : deliverTokenData()
SalesforceCRM --> Customer : sendCustomerNotification()
Customer --> AEMFrontendApp : submitCorrection()
@endum1
```

# Key Enhancements

- Enums and domain classifications added for critical fields like status, priority, severity
- Access auditing (e.g., lastAccessed, accessCount, requestIp)
- Temporal tracking: createdAt, updatedAt, lastSyncedAt
- Context-rich attributes: like region, assignedAgent, submittedCorrections

Would you like to **generate a PlantUML image** from this? Or proceed to the **sequence diagram for a specific flow** (e.g., error generation, token lookup, or Excel sync)?

Du:

can you please make it look nicer so that it is more compact together and below each other etc., because it looks very wild at the moment

ChatGPT

To make your UML class diagram more compact and visually structured, we'll apply layout management techniques:

## Optimizations Applied

- 1. Logical vertical grouping with together {} to cluster:
  - o Internal domain services
  - Infrastructure classes
  - External systems
- 2. LAYOUT\_TOP\_DOWN() to force a clean top-to-bottom structure
- 3. Removed unnecessary horizontal sprawl by clustering class groups
- 4. Kept all relationships intact and readable

# **<sup>®</sup> Cleaned-Up & Structured UML Class Diagram (PlantUML)**

```
+ String generateToken() <<internal>>
      boolean validateErrorPayload() <<internal>>
  }
  class UrlBuilderService <<Azure Logic App>> {
    - String token
      String countryCode
    - String baseUrl
    - String countryUrl
    + String buildUrl() <<internal>>
  }
  class ErrorDataCollector <<Azure Logic App>> {
    - String token
      String countryUrl
      List<String> errorCodes
    - Map<String, String> metadata
    - Map(String, String, mecanata

- String ruleId

- String severity <<Enum { info, warning, critical }>>

- String resolutionStatus <<Enum { unresolved, pending, resolved }>>

- String errorCategory <<Enum { format, logic, missingData }>>
    + void enrichWithLocalizationData()
    + void storeErrorRecord()
      void prepareCasePayload()
      void forwardToSalesforce() <<HTTP POST>>
  class LocalizationSyncService <<Azure Logic App>> {
    String fileNameDateTime lastModified
    - String updatedBy
      void monitorExcelUpdates() <<Event Grid>>
      void syncDataToTableStorage()
  class ExcelAutomationScript <<Excel Script (private)>> {
    String localPathDate lastUpload
    + void uploadLocalizationFile() <<HTTPS PUT>>
  class SalesforceIntegrationService <<Azure Logic App>> {
    - String caseId
    - Map<String,String> structuredPayload
    String syncStatus <<fnum { pending, synced, failed }>>DateTime lastSyncedAt
    - String assignedAgent
    - String priority <<Enum { low, medium, high }>>
    + void pushToSalesforce() <<HTTP POST>>
  }
  class TokenQueryApi <<Azure Function (public)>> {
    String requestIpDateTime lastAccessed
    - Integer accessCount
    + Map<String, String> getErrorDataByToken(String token) <<HTTP GET>>
 }
 INFRASTRUCTURE SERVICES
together {
 class AzureServiceBus <<Enterprise Service Bus (internal)>> {
    + void sendMessage(Object payload)
    + Object receiveMessage()
  class ErrorDataStore <<Azure Table Storage (internal)>> {
    - String partitionKey
    - String rowKey
    - String token
    - String countryUrl
    List<String> errorCodesMap<String, String> metadata
    - DateTime createdAt
    - DateTime updatedAt
    + void insertRecord()
+ Map<String, String> fetchRecordByToken(String token)
 class LocalizationBlobStore <<Azure Blob Storage (internal)>> {
    - String blobFile
    - String uploaderId
    - String region
    + String readExcelFile()
+ void writeExcelFile()
 EXTERNAL SYSTEMS
together {
```

```
class OrderAnalysisEngine <<Order Processing (external)>> {
       void sendErrorPayload() <<HTTPS POST>>
  class AEMFrontendApp <<React SPA (external)>> {
     - String formRedirectUrl
       boolean feedbackProvided
     - Integer submittedCorrections
    + void requestTokenData(String token) <<HTTPS GET>>
       void renderErrorDetails()
       void redirectToCorrectionForm()
  }
  class SalesforceCRM <<CRM (external)>> {
     + void receiveCase() <<HTTPS POST>>
       void sendCustomerNotification() <<Email>>
  }
  class Customer <<End User (external)>> {
    - String sessionId
- String email
    + void uploadDocument()
    + void viewErrorDetails()
     + void submitCorrection()
  }
  _____
  RELATIONSHIPS
OrderAnalysisEngine --> AzureServiceBus : sendErrorPayload()
AzureServiceBus --> TokenService : receiveMessage()
TokenService --> UrlBuilderService : generateToken()
UrlBuilderService --> ErrorDataCollector : buildUrl()
ErrorDataCollector --> LocalizationBlobStore : enrichWithLocalizationData()
ErrorDataCollector --> ErrorDataStore : storeErrorRecord()
ErrorDataCollector --> SalesforceIntegrationService : prepareCasePayload()
SalesforceIntegrationService --> AzureServiceBus : pushToSalesforce()
AzureServiceBus --> SalesforceCRM : receiveCase()
ExcelAutomationScript --> LocalizationSyncService : uploadLocalizationFile()
LocalizationSyncService --> LocalizationBlobStore : monitorExcelUpdates()
LocalizationSyncService --> ErrorDataStore : syncDataToTableStorage()
AEMFrontendApp --> AzureServiceBus : requestTokenData()
AzureServiceBus --> TokenQueryApi : getErrorDataByToken()
TokenQueryApi --> ErrorDataStore : fetchRecordByToken()
TokenQueryApi --> AzureServiceBus : returnResponse()
AzureServiceBus --> AEMFrontendApp : deliverTokenData()
SalesforceCRM --> Customer : sendCustomerNotification()
Customer --> AEMFrontendApp : submitCorrection()
@enduml
```

Let me know if you want me to:

- Generate a PDF or image output
- Split this diagram into subdomains
- Provide sequence diagrams or deployment layout next.





Dateien auswählen Keine Datei ausgewählt Dateien auswählen Keine Datei ausgewählt