Breanos Intelligence Industry Assistant A.D.A.M.



Neutorstraße 13  
5020 Salzburg

AUSTRIA

Tel: +43 (662) 276198-11

Fax: +43 (662) 276198-98

Mail: [office@breanos.com](mailto:office@breanos.com)

|  |  |
| --- | --- |
| Version | 1.0 |
| Datum | 22.05.2018 |

Inhaltsverzeichnis

[1 Änderungsverzeichnis 4](#_Toc514759714)

[2 Breanos Intelligence Industry Assistant A.D.A.M. 5](#_Toc514759715)

[2.1 Daipan.Assistent.Adam.Core 5](#_Toc514759716)

[2.2 Daipan.Assistent.Adam.Presenter – Präsentation 6](#_Toc514759717)

[2.3 Daipan.Assistent.Adam.Client 6](#_Toc514759718)

[2.4 Daipan.Assistent.Adam.EngineGroup 6](#_Toc514759719)

[2.5 Daipan.Assistent.Adam.EngineGroup.WorkerEngine 7](#_Toc514759720)

[2.6 Daipan.Assistent.Adam.Session 7](#_Toc514759721)

[2.7 Daipan.Assistent.Adam.Security 7](#_Toc514759722)

[2.8 Daipan.Assistent.Adam.Communication – Externe Kommunikation 8](#_Toc514759723)

[2.9 Daipan.Assistent.Adam.Communication.Blackboard 8](#_Toc514759724)

[2.10 Daipan.Assistent.Adam.AccessManagement/Control – Client Zugriffsverwaltung 8](#_Toc514759725)

[2.11 Daipan.Assistent.Adam.AIEngine 8](#_Toc514759726)

[2.12 Daipan.Assistent.Adam.ExternalCommunication ECOM 8](#_Toc514759727)

[2.13 Komponententype 10](#_Toc514759728)

[2.14 Daipan.Assistent.Adam.SfaKpuHost 10](#_Toc514759729)

[2.15 Datenfluss Von der KPU zum Blackboard 10](#_Toc514759730)

# Änderungsverzeichnis

|  |  |  |  |
| --- | --- | --- | --- |
| Datum | Version | Verfasser | Beschreibung |
| 2018-05-22 | 1.0 | EBE | Zusammengefügte Version. |

# 2 Breanos Intelligence Industry Assistant A.D.A.M.

Der Breanos Intelligence Industry Assistant besteht aus mehreren Komponenten die nachfolgend beschrieben werden. Jeder dieser Komponenten beinhaltet für die Arbeitsweise des Assistent essenzielle Funktionalität.

Programmatisch werden die einzelnen Komponenten ein Namensbereiche sprich Namespaces gegliedert.

Die 13 von uns identifizierten Komponenten bzw. Namespaces sind folgendermaßen gegliedert:

|  |  |
| --- | --- |
| **Komponentenname** | **Beschreibung** |
| Daipan.Assistent.Adam.Core | Den Presenter laden. Übersicht über alle ihm zugeordneten laufenden EngineGroups beschaffen. |
| Daipan.Assistent.Adam.Presenter | Präsentationsschicht. Hat Liste von Clients. |
| Daipan.Assistent.Adam.Client | Client dient zur Visualisierung und als Controller. |
| Daipan.Assistent.Adam.EngineGroup | Repräsentiert eine Produktionslinie |
| Daipan.Assistent.Adam.EngineGroup.WorkerEngine | Repräsentiert eine Arbeitseinheit |
| Daipan.Assistent.Adam.Session | Beinhaltet alle Clients und EngineGroups |
| Daipan.Assistent.Adam.Security | Client Security |
| Daipan.Assistent.Adam.Communication | Externe Kommunikation |
| Daipan.Assistent.Adam.Communication.Blackboard | Kommunikation zum Blackboard |
| Daipan.Assistent.Adam.AccessManagement/Control | Zugriffsverwaltung der einzelnen Anwendungen |
| Daipan.Assistent.Adam.AIEngine | AI Auswertung/Selbstständige Tätigkeit |
| Daipan.Assistent.Adam.ExternalManagement | Messaging System |
| Daipan.Assistent.Adam.SfaKpuHost | Service zum Instanziieren von KPUs. |

Tabelle 1

## Daipan.Assistent.Adam.Core

Als Service zu implementieren. Wird auch beim Hochlaufen eine wichtige Rolle spielen.

Hauptkomponente/Mainkomponente:

Alle XAML-Files kommen in eine gemeinsame Datenbank.

Die Core Komponente hat die Aufgabe den Presenter zu laden. Weiters muss Sie sich einen Überblick über alle laufenden **EngineGroups** beschaffen.

Alle EngineGroups und WorkerEngines werden in WorkerDefinition.xml abgelegt. Die XML-Datei wird als BLOB in einer Datenbank hinterlegt.

Legt eine Liste von Standard KPUs an. Diese ist über Core erreichbar. Diese KPUs können abgefragt, gestartet oder gestoppt werden.

Neben den Standard KPUs gibt es auch eine Liste von MU-KPUs diese werden beim Hochlauf geladen.

Im Core wird eine Liste von Sessions gehalten.

## Daipan.Assistent.Adam.Presenter – Präsentation

Der Präsenter wird von der Core Komponente instanziiert. Der Präsenter hat eine Liste von Clients.

Die Clients werden je nach Notwendigkeit vom Presenter instanziert. Client Instanzen sind Actoren.

Presenter – API.

## Daipan.Assistent.Adam.Client

Jeder Client enthält einen Controller. Das Prinzip in dieser Komponente erfolgt nach dem Model View Controller Prinzip. D.h. im Model werden die visuellen Daten angezeigt. Der Controller legt die Berechtigungsebene fest bzw. stellt Anfragen an das **AccessManagement**. Model sind die Daten WorkerGroup, WorkerEngine.

Jeder Client kann zu einem Zeitpunkt nur eine Referenz auf eine **WorkerEngine** haben.

In der Client-Komponente ist das XAML abgelegt.

## Daipan.Assistent.Adam.EngineGroup

Eine **EngineGroup** repräsentiert eine Halle bzw. Produkionsstandort.

Engine Groups werden in der Datei **EngineGroupDefinition.xml** festgelegt. Diese ist in Form einer XML Datei strukturiert.

<?xml version="1.0"?>

<EngineGroups>

<EngineGroup name="Group\_1">

<WorkerEngine name="Bohren\_1” type=”h3”/>

<WorkerEngine name="Fräsen\_1" type=”f5”/>

<WorkerEngine name="Fräsen\_2" type=”f5”/>

</EngineGroup>

<EngineGroup name="Group\_2">

<WorkerEngine name="Bohren\_1” type=”h3”/>

<WorkerEngine name="Bohren\_2” type=”h3”/>

</EngineGroup>

</EngineGroups>

## Daipan.Assistent.Adam.EngineGroup.WorkerEngine

* Definitionsfile: WorkDefinition. (XAML/XML).

Eine WorkerEngine repräsentiert eine Produktionseinheit. Sie kann den Zustand:

* + Running Wird gerade ausgeführt.
  + ToDo Ist geplant, läuft noch nicht.
  + Ready Auftrag verarbeitet.

In der WorkerEngine wird die PLC-KPU gekapselt bzw. referenziert.

Arten der KPUs:

* MU Maschine Unit

Repräsentiert eine KPU für eine konkrete Maschine.

* Logistics

Organisiert die KPUs untereinander. Z.B. Just in Time delivery. Eine Logistics KPU fordert eine Menge von Rohteilen an. Diese versucht dann zu gewährleisten, dass diese geliefert bzw. produziert werden.

* Transport Unit

KPU für die Transportwege. (Förderbänder).

XAML-Definition wird im WorkerEngine gespeichert.

Zur Konfiguration der WorkerEngine gehören dabei zwei definierende XML-Dateien

* Die „ProductionDefinition“. Sie stellt einen konkreten Ablauf zur Produktion eines Produkts in der Fabrik dar. Das Produkt steht im Mittelpunkt.
* Die „FactoryDefinition“. Sie stellt die Fabrik als ganzes dar, mit allen darin befindlichen, über KPUs abgebildeten Maschinen, Transportwegen, etc.

Eine KPU ist immer für eine Maschine zuständig. Sie kann aber auch für mehrere zuständig sein.

## Daipan.Assistent.Adam.Session

* Client Collection
* EngineGroup Collection

Im Core wird eine Liste von Sessions gespeichert. An einer Session hängen 1 oder mehrere **WorkerEngines**.

Es existiert immer eine Session pro Viewer. Eine große View das die aktuelle View des Viewers ist. Die große View kommt von der WorkerGroup.

Im großen View der WorkGroup werden die Viewes der WorkerEngines eingeblendet.

Zu einer View gehören nach dem Model View Controller Prinzip auch ein Controller und ein Model.

## Daipan.Assistent.Adam.Security

* Client, ClientAssistent

Dient zur verschlüsselten Kommunikation mit den außenstehenden Clients. Weiters muss auch vor Replay Attacken abgesichert werden.

In der ersten Version des Assistent reicht es wenn die Verschlüsselung mit https erfolgt.

## Daipan.Assistent.Adam.Communication – Externe Kommunikation

* Externe Kommunikation zu anderen Assistents.

Die Communication Komponente dient als externe Kommunikationszentralle. Sie beinhaltet Schnittstellen zu:

* anderen Assistenten.

Als Beispiel sei folgendes Beispiel erwähnt. Zwei Produktionsstandorte die geographisch voneinander getrennt sind. Die Produktion muss beispielsweise für eine bestimmte Anzahl von Produkten Just in time gestartet werden. So muss der Assistent vom Produktionsstandort 1 dem Assistent vom Produktionsstandort 2 eine Message schicken um die Produktion zu starten.

Die Kommunikation zwischen den Knoten wird über SignalR erfolgen. Alle Assistent Instanzen sind vollwertige Assistents. D.h. Sie beinhalten

* (Third party Anwendungen.)

## Daipan.Assistent.Adam.Communication.Blackboard

* Kommunikation zum/vom Blackboard. In diesem Namensbereich wird die Funktionalität des Blackboards integriert. Das Blackboard dient als zentrale Kommunikationszentrale für die Applikation.

## Daipan.Assistent.Adam.AccessManagement/Control – Client Zugriffsverwaltung

Zugriffsverwaltung/Berechtigungsmanagement der einzelnen Anwendungen. Wird auch für den Graphical Assistent verwendet. Darin wird definiert welche Halle dargestellt wird. Z.b Geely oder SiFang. Wenn drei Viewer an den selben Client interessiert sind, wird jeweils nur auf den ersten Client verwiesen.

## Daipan.Assistent.Adam.AIEngine

Ist die Komponente die Benachrichtigungen per Voice/Gedanken oder per Eingabegerät wie Datenhandschuh entgegennimmt.

Die AI führt Auswertungen durch und/oder verschickt Benachrichtigungen und/oder führt selbständig Tätigkeiten durchführt.

## Daipan.Assistent.Adam.ExternalCommunication ECOM

(ExternalManagement)-Message System, Benachrichtigungssystem für die Zugriffsverwaltung für die externen Nachrichten.

Anbindung an dem Assistenten beispielsweise über TCP/IP, Service Bus etc.

External Management ist für die Kommunikation mit den Viewern zuständig. Zu diesen wird eine http/REST Verbindung aufgebaut.



Abbildung 1Diagram

## Komponententypen

|  |  |
| --- | --- |
| **Component Name** | **Service Type** |
| Core | Service |
| Presenter | Service |
| Client | Actor |
| EngineGroup | Actor |
| WorkerEngine | Actor |
| Session | Actor |
| Security | Service |
| Communication/ICOM Inter | Service |
| BlackBoard | Actor? |
| AccessManagement/Control | Service |
| AIEngine | Actor |
| ExternalManagement/ECOM | Service |
| SFAKpuHost | Service |

## Daipan.Assistent.Adam.SfaKpuHost

Sfa Service zum Instanzieren von KPUs.

## Datenfluss Von der KPU zum Blackboard

Siehe UML Diagramm in Enterprise Architect:

Breanos.Daipan.Biks.Architecture.Blackboard.Black Board Implementation

## Methodendefinitionen für den View Client

**Client Methodes**

|  |  |
| --- | --- |
| OnReceiveBadLogin() |  |
| OnAddToMasterList(string text, string id) |  |
| OnRemoveFromMasterList(string id) |  |
| OnMessageReceived(string title, string message, List<string> buttons) | might need more clarification |
| OnDataReceived(string viewId, string itemId, string value) |  |
| OnViewReceived(string id, string xaml) |  |
| OnFileReceived(string relativePath, byte[] data) |  |
| OnViewRequirementsReceived(string viewid, string[] requirements) |  |

**Server methods**

|  |  |
| --- | --- |
| Handshake(string locale) | might need more parameters / structured meta info about client |
| Login(string user, string password, string domain) |  |
| RequestExecute(string functionId, List<string> parameters) |  |
| RequestViewRequirements (viewId) |  |
| RequestFile(string fileReference) | ? relative path? id? |