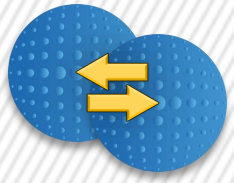
OMNIS-ZU-OMNIS BIDIREKTIONALE VERKNÜPFUNG

UND

ASYNCHRONES CONTENT MANAGEMENT

Francesco Del Dotto - fdotto@softpi.com



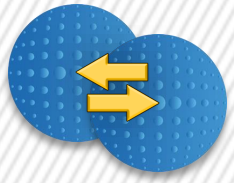
Wann ist eine **Omnis Studio** Instanz einfach nicht genug?

Drei Situationen, die diese Frage beantworten:

1. Lange und Komplexe, Tasks mit hoher Last, die in JS Client-Sessions ausgeführt werden

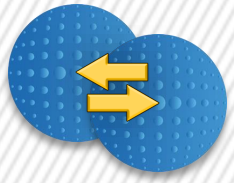
2. Ein zentraler Mehrzweckdienst, der mit Omnis Studio entwickelt wurde und auf den viele Einzweck-Serveranwendungen aus der Ferne zugreifen müssen

3. Zwei oder mehr Omnis Web Server-Instanzen, die miteinander kommunizieren, aber in getrennten Umgebungen bleiben müssen



Lange und komplexe Tasks

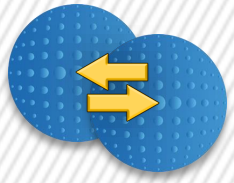
- Riesige PDF-Ausdrucke, die umfangreiche Berechnungen, Bildverwaltung oder Aufgaben mit großen Dateien erfordern, können den Arbeitsablauf des Benutzers verlangsamen
- Ein Benutzer, der zeitaufwändigere Aufgaben an einen "Geisterbenutzer" weiterleiten kann, der seine Arbeit erledigt, kann auf der Hauptinstanz arbeiten und auf Ergebnisse warten



Remote-Zugriff auf einen Multi-Purpose Omnis Web Server

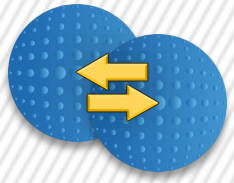
- Wenn ein Satz allgemeiner Aufgaben und Methoden in mehreren Bibliotheken verwendet wird, müssen wir, um sie zu aktualisieren, jede einzelne von ihnen neu bereitstellen.
  Sind sie in einer eigenen Instanz enthalten, kann ihre Bereitstellung nur einmal erfolgen

- Es ist möglich, eine modulare Reihe von Omnis App Servern zu erstellen, die in unterschiedlichem Umfang zusammenarbeiten, wobei jeder Instanz eine bestimmte Bezeichnung zugewiesen wird, im Gegensatz zur Lastverteilung.



Interaktion zwischen Omnis-Instanzen

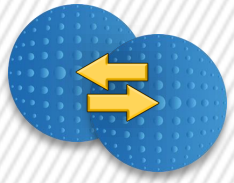
- Wenn mehrere Instanzen miteinander kommunizieren können, ist es möglich, ein System zu schaffen, bei dem eine asynchrone und direkte Interaktion erforderlich ist.
  Ein Beispiel dafür wäre ein Text-Chat oder ein anderes Nachrichtensystem, das nativ in eine Bibliothek integriert ist und benutzerdefinierte Interaktionen mit dem Bibliothekscode ermöglicht.
- Es kann in Situationen helfen, in denen die Benutzerbibliothek in einer Sandbox untergebracht werden muss und direkt auf Daten aus anderen Instanzen zugegriffen werden muss
- Es kann Bibliotheken den Zugriff auf andere Betriebssysteme und ihre exklusiven Funktionen ermöglichen, indem eine "Zwillingsinstanz" an anderer Stelle installiert wird.



Wie funktioniert das?

Die **Omnis Interface Reference** hat drei grundlegende Funktionen:

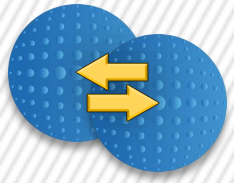
- Variable GET
- Variable SET
- Methodenausführung



Wie funktioniert das?

Es kann Bibliotheken den Zugriff auf andere Betriebssysteme und ihre exklusiven Funktionen ermöglichen, indem eine "Zwillingsinstanz" an anderer Stelle installiert wird. :

1. Ein HTTP OW3 Worker, der die POST Requests, die die Daten enthalten, sendet
2. Ein RESTful Service, der die Anfragen empfängt, eine grundlegende Übersetzung der Nachrichten vornimmt und das Object Interface aufruft
3. Ein Object Interface, das die GET/SET- und Ausführungsanforderungen sammelt und beim Senden von Daten für die Übertragung vorbereitet und beim Empfangen von Daten den Synchronisationscode ausführt

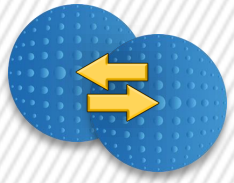


Wie funktioniert das?

Die OIR-Funktionen werden mit einem dreiteiligen Framework verwaltet, das auf jeder Seite eingesetzt wird:

- GET
- SET
- Method

Die Anfrage wird an das OIR-Objekt gesendet, das den HTTP-Worker verwendet, um den RESTful-Dienst aufzurufen



Wie funktioniert das?

Die OIR-Funktionen werden mit einem dreiteiligen Framework verwaltet, das auf jeder Seite eingesetzt wird :

- GET
- SET
- Method

- Form $ident
- Task $ident
- Variablenwert
- Methodenparameter

Die Anfrage wird an das OIR-Objekt gesendet, das den HTTP-Worker verwendet, um den RESTful-Dienst aufzurufen

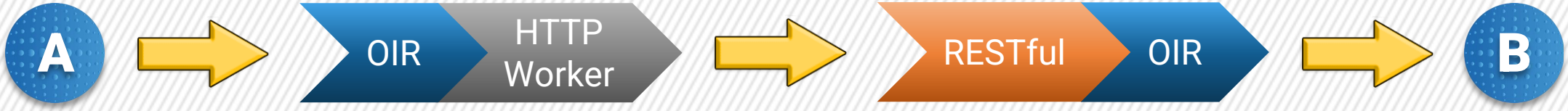
Absender-“Adresse" wird an die andere Instanz gesendet, um zurückzu-verfolgen

Wie funktioniert das?

Die OIR-Funktionen werden mit einem dreiteiligen Framework verwaltet, das auf jeder Seite eingesetzt wird:

- GET
- SET
- Method

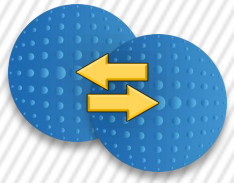
- Form $ident
- Task $ident
- Variablenwert
- Methodenparameter



Die Anfrage wird an das OIR-Objekt gesendet, das den HTTP-Worker verwendet, um den RESTful-Dienst aufzurufen

Absender-“Adresse" wird an die andere Instanz gesendet, um zurückzu-verfolgen

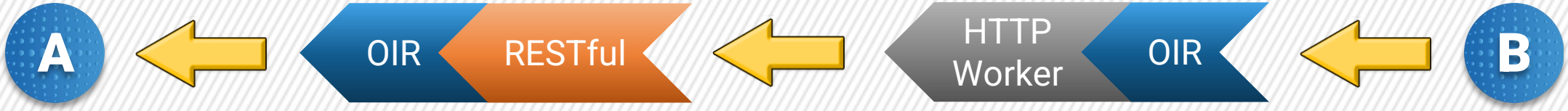
Das empfangende OIR speichert die Daten der Variablen in einer Startup_Task-Liste zusammen mit ihrer ursprünglichen „Adresse" oder führt die Anforderungsmethode von einer Objekt- oder Codeklasse aus



Wie funktioniert das?

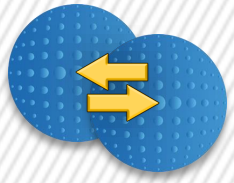
Es funktioniert ach rückwärts!

- Ursprüngl. Form $ident
- Ursprüngl. Task $ident
- Neuer Variablenwert
- Methodenparameter



Das empfangende OIR setzt eine Aufgaben- oder Instanzvariable, die zum ursprünglichen Formular oder zur ursprünglichen Aufgabe gehört

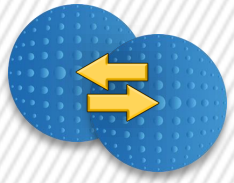
Das OIR fordert ein SET einer geänderten Variablen oder eine Rückgabemethode, die einen Download auf einem Client des ersten Web Servers startet



Wie funktioniert das?

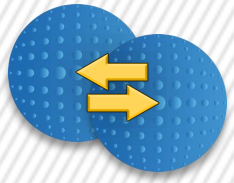
Was genau läuft im Core?

- Die SET-Funktion stützt sich auf eine Task-Liste auf der Empfangsseite, in der die Ursprungsaufgabe, der Name des Ursprungsformulars und der Wert einer im sendenden Web Server enthaltenen Variablen gespeichert sind
- Die Werte werden als Spalten des Typs kRow gespeichert, so dass jede Art von Daten darin gespeichert werden kann.
- Eine Methode kann unter Verwendung von zuvor festgelegten Variablen und/oder durch Übergabe von Parametern an sie ausgeführt werden.
- Parameter und Variablenwerte werden als JSON-Inhalt übergeben. Aufgabenkennung, Formularkennung, Variableninfo und Methodennamen werden als Kopfdaten übergeben



Wie sieht es mit der Sicherheit aus?

- Die Verbindung basiert auf RESTful Services und HTTP-Workern, so dass jede Verbindung mit Omnis-Code, interner Verschlüsselung oder jeder anderen Methode zum Schutz von RESTful Web Services verwaltet und gefiltert werden kann.
- Der äußere Zugriff auf die Daten beschränkt sich auf das Schreiben von Werten innerhalb der „Master“-Task-Liste. Der Code kann unter Berücksichtigung dieses Aspekts geschrieben werden, um unerwünschte Inhalte zu vermeiden.



Können wir das auch nativ bekommen?

Der Aufbau einer Omnis Instance Reference kann mühsam sein, besonders am Anfang. Eine Form des Zugriffs auf andere Instanzen zu haben, selbst wenn diese auf RESTful und HTTP OW3 Workers basieren (Technologien, die von Omnis Studio bereits nativ unterstützt werden), würde dies deutlich vereinfachen.

Diese Art von Fähigkeit in einem gewissen nativen Umfang wäre zumindest für den Zugriff auf Variablen oder den Aufruf von Methoden in ähnlicher Weise wie bei OLE Automation nützlich.

Wenn dies umgesetzt werden könnte, könnte es eine Lösung sein, die zu einer weiteren Ebene in der Omnis-Programmierung führt, die eine Verbindungsfähigkeit und Flexibilität in der Load- und Aufgabenverwaltung ermöglicht, die bisher schwer zu erreichen ist.

Vielen Dank!

Francesco Del Dotto - fdotto@softpi.com