



The Retail Innovators

SAP Dynamic Pricing by GK

Architecture Guide

Version: v4.0.0



COPYRIGHT

© 2022 SAP SE or an SAP affiliate company. All rights reserved. No part of this publication may be reproduced or transmitted in any form or for any purpose without the express permission of SAP SE or an SAP affiliate company.

1. You may not use the SAP Material for a purpose competitive with SAP or its products unless otherwise clearly permitted by applicable law.
2. You may not use the SAP corporate logo.
3. No use of other SAP trademarks is granted under this section. For information regarding use of SAP trademarks, see <http://www.sap.com/corporate/en/legal/trademark.html>.

SAP and other SAP products and services mentioned herein as well as their respective logos are trademarks or registered trademarks of SAP SE (or an SAP affiliate company) in Germany and other countries. All other product and service names mentioned are the trademarks of their respective companies.

Internal Document Information: 1222052053 | 2022-02-28

TABLE OF CONTENTS

1	Overview	4
2	Landscape	5
2.1	Landscape Overview.....	5
2.1.1	Dynamic Pricing Service.....	5
2.1.2	Dynamic Pricing Web Client.....	6
2.1.3	Database.....	6
2.2	Data Integration.....	6
3	Separation of Data	6
3.1	Overview.....	6
3.2	Tenants and Business Units.....	7
3.3	Business Units.....	7
4	Technical Architecture	9
4.1	Overview.....	9
4.1.1	Linux, Java, Tomcat.....	9
4.1.2	AIR Plugin Runtime.....	9
4.1.3	Plugins: Pricing, Pricing AI, Storage.....	9
4.1.4	Browser.....	9
4.1.5	SQL Database.....	9
4.2	AIR Runtime Services.....	10
5	Deployment Options	11
5.1	Cloud Deployment.....	11
5.1.1	Cloud Deployment with Backend using REST.....	11
5.1.2	Cloud Deployment with SAP Backend (on Premises).....	11
5.2	On Premises Deployment.....	12
6	Scaling Options	12
7	High Availability	13
7.1	High Availability of Cloud Services.....	13
7.2	High Availability Configuration of the Application.....	13
8	Software Development Kit	14
8.1	Overview.....	14
8.2	Integration of Custom Code.....	14

1 Overview

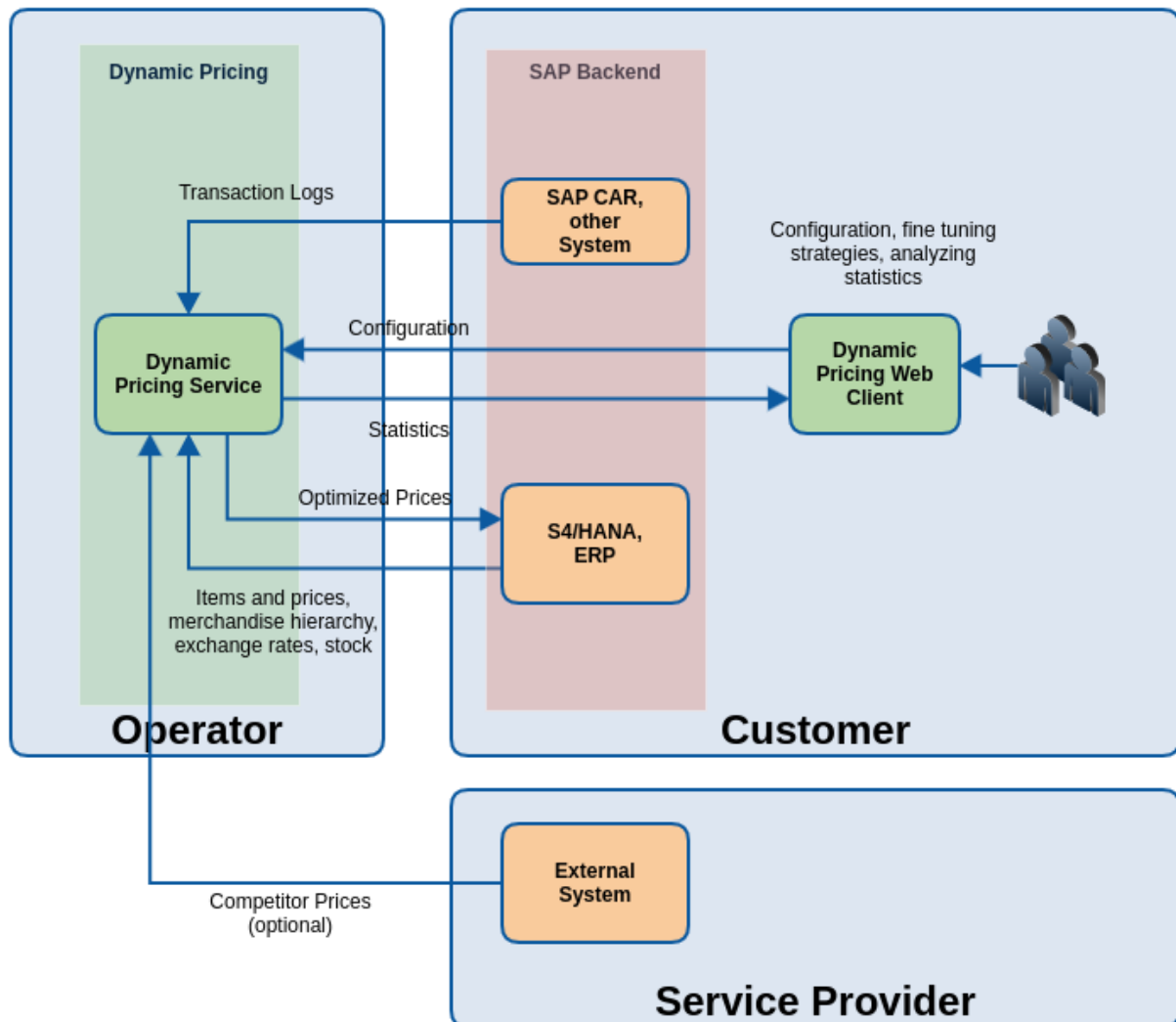
This document provides architectural information. The document describes the following:

- Dynamic pricing landscape
- Data integration options
- Separation of data
- Overview of the technical architecture
- Deployment options
- Scaling
- Overview of extension options via SDK

If you are looking for details about the sizing, please refer to the "Sizing Guide". The Software Development Kit (SDK) is detailed in the "Extension Guide".

2 Landscape

2.1 Landscape Overview



2.1.1 Dynamic Pricing Service

The dynamic pricing service imports master data from an ERP system to get information about items and their prices, merchandise hierarchy and assignment of items to the categories. The item data can be enriched with additional item data that is relevant to analyze the customer purchase behavior. The hierarchy category can be used to partition the item assortment into one or more item segments to fine-tune different price optimization strategies for each segment. Some price optimization strategies take competitor prices into account. In that case, also competitor prices should be imported into the system. If the pricing service has to convert currencies (import and/or export), then you should also supply it with up-to-date exchange rates. All master data is persisted in the database.

The pricing service regularly computes the optimized prices and persists them in the database. The computation logic is encapsulated in so-called price strategies. A price manager may fine-tune the configuration of those price strategies, based on the current statistics and/or simulation runs that compare different price strategy configurations.

Finally, the optimized prices are exported to other systems, such as ERP.

2.1.2 Dynamic Pricing Web Client

The Web Client offers a rich user interface to configure:

- Master data import, including scheduling
- Item segments and assignment of price strategies for each segment
- Scheduled price optimization runs
- Optimized price export, including scheduling
- User management with users and roles
- Management of business units

The pricing service offers a broad REST API to access the service. This is mostly used for export and import.

2.1.3 Database

For persistency of master data, configuration and optimized prices, a SQL database is used.

2.2 Data Integration

The dynamic pricing service offers various interfaces for integration, for importing and exporting data. There is a set of product standard interfaces that can be extended. It is also possible to develop custom import and export interfaces with the SDK.

For a detailed list of product standard interfaces and their configuration, please refer to the documents "Integration Guide" and "Configuration Guide". The following table just outlines the interface types:

Direction	Type of Data
Import	Items (price, attributes), merchandise groups, merchandise hierarchy, merchandise group assignment
Import	Item price limits
Import	Transaction log
Import	Competitor prices
Import	Exchange rates for currencies
Import	Stock
Export	Optimized prices

3 Separation of Data

3.1 Overview

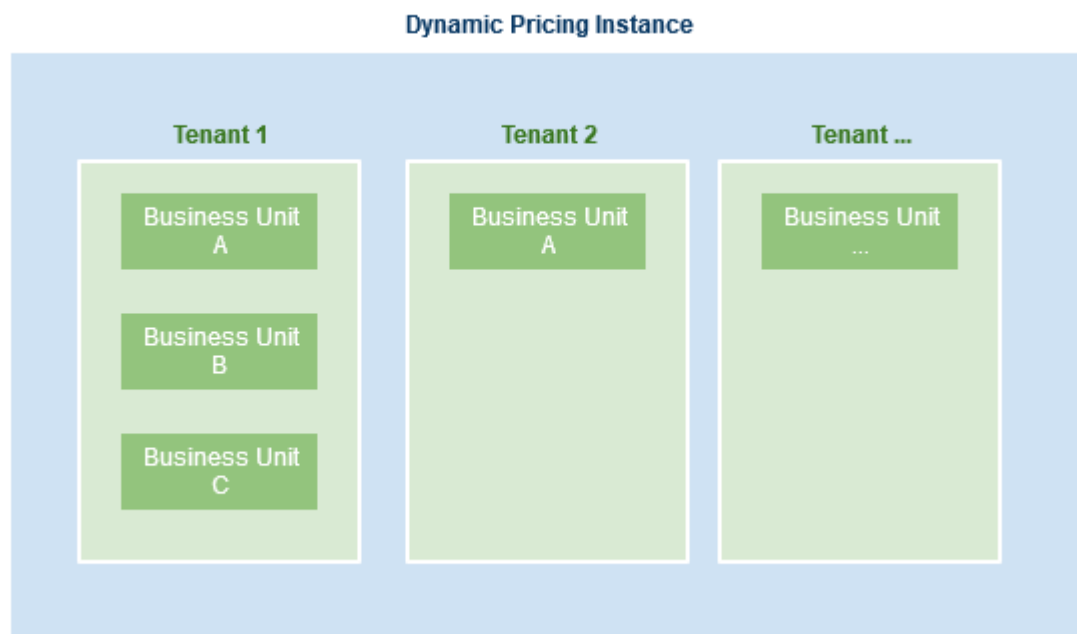
The dynamic pricing service offers these concepts for separation of data:

Separation Concept	Database Separation	Collaboration	API Separation	Software Version	Data Access	Description	Example
Tenant	Database	No	URL	Can vary between tenants	Users and roles defined for tenant Data of other tenants cannot be accessed.	All business units belong to one tenant. Tenants are strictly separated from each other and do not see each others data. Own set of users and roles for each tenant.	
Business Unit	Business Unit database table column in all tables that separates the data.	Yes	URL	Tenant software version is used for all business units	Permissions to access business unit Tenant permissions can be restricted to business unit. Data of other business units can be accessed via defined business processes.	Business units have their own data but can collaborate with other business units.	Web Shop, Store 1, 2, 3

3.2 Tenants and Business Units

A tenant separates all data from other tenants. Tenants are implemented as separate databases with own schemas and access rights. In the software runtime, it is modeled as instances that can host different sets of software plugins.

A tenant is made up of at least one business unit. A business unit can be used to model different countries or retail shops. Each business unit can have its own master data, transactions and independent price optimization strategies. Each can run its own importers and export data independently. There are some well-defined scenarios to link business units together.

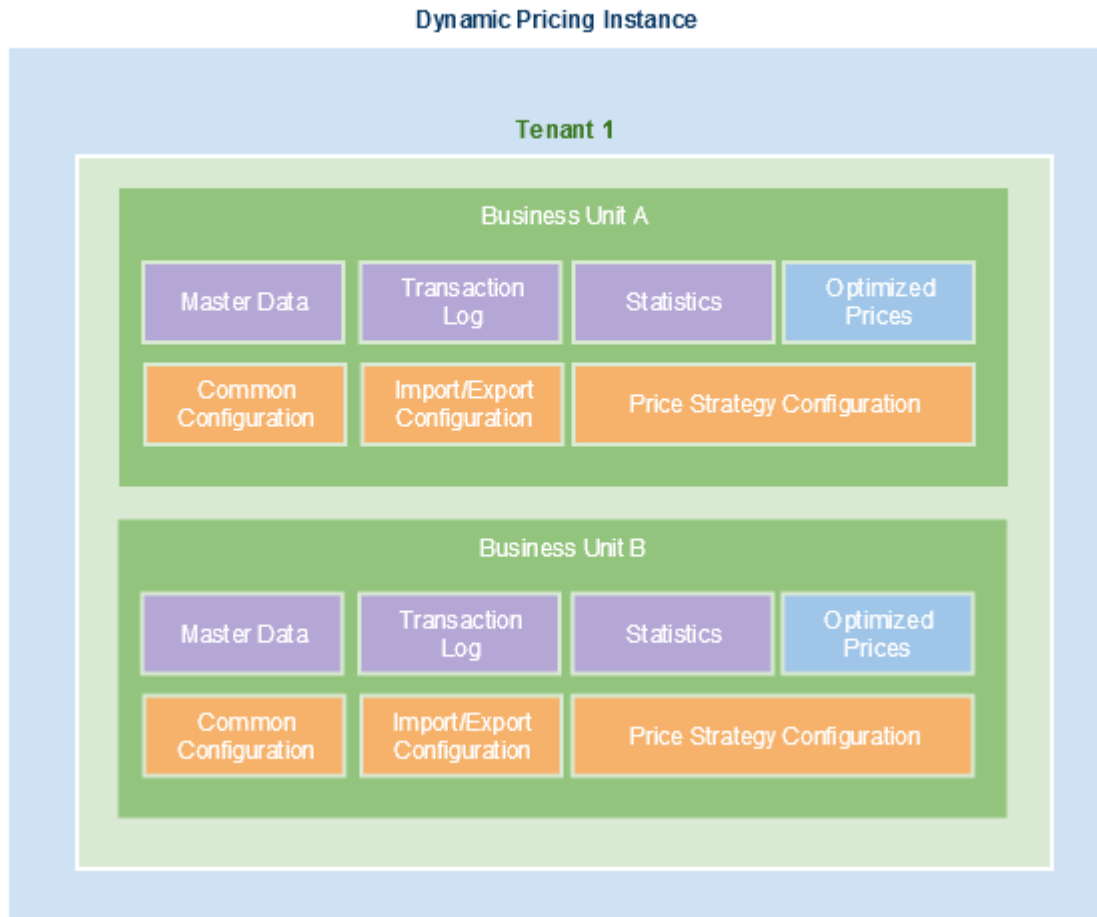


3.3 Business Units

Each business unit has its own set of:

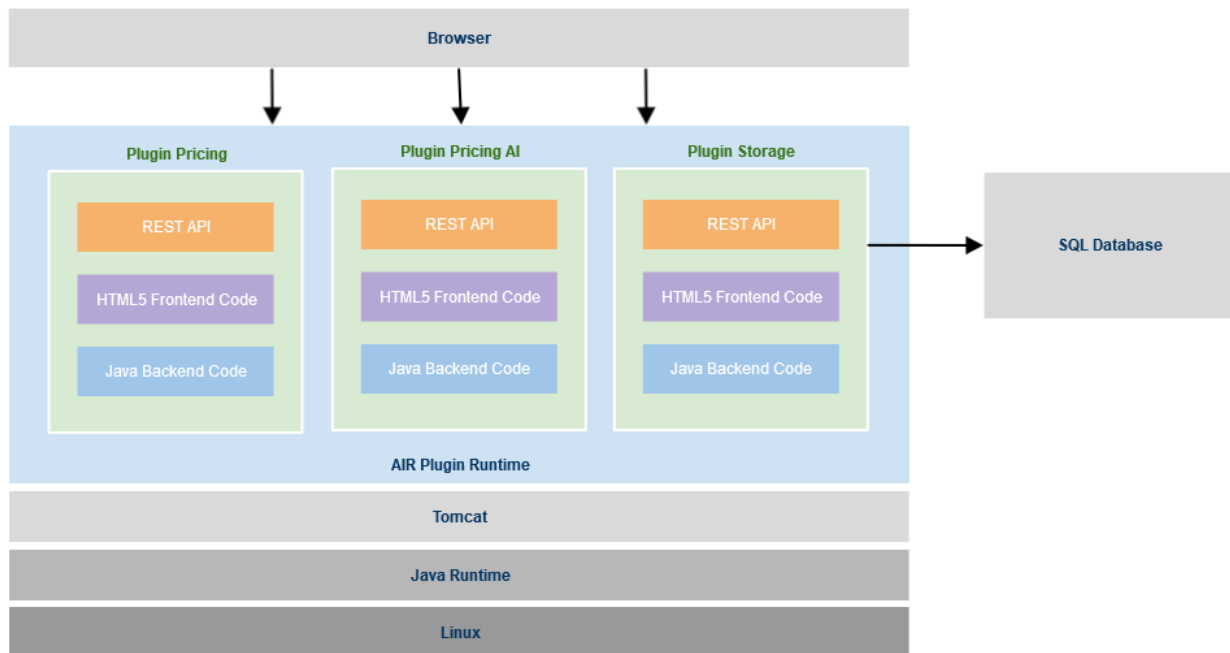
- Price strategy configuration
- Common configuration (e.g. currency)

- Master data
- Transactions
- Statistics
- As a result: optimized prices



4 Technical Architecture

4.1 Overview



4.1.1 Linux, Java, Tomcat

Operating system for running the Java Runtime (JDK). The Tomcat container provides resource pooling and the servlet implementation.

4.1.2 AIR Plugin Runtime

A runtime for plugins. Plugins offer services. The architecture supports services from microservices up to large applications. The runtime offers many services like security, messaging or a common portal to register. See next chapter for more details.

4.1.3 Plugins: Pricing, Pricing AI, Storage

The application plugins. The standard product "Dynamic Pricing" is made up of the pricing and the storage plugins. Besides that, the pricing AI (artificial intelligence) plugin is shipped containing a set of standard price strategies. The pricing AI plugin is an extension plugin. With the SDK you can create custom extension plugins with custom price strategies and/or custom importers, exporters and user interface extensions.

4.1.4 Browser

An HTML5 capable browser (see the PAM for details about browser types and versions) that renders the user interface.

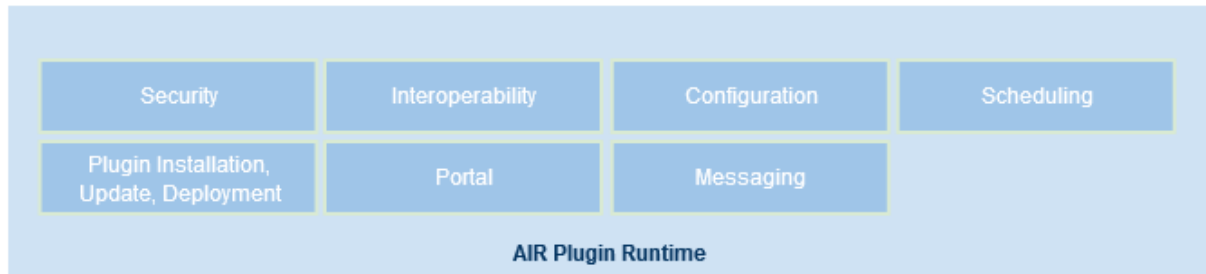
4.1.5 SQL Database

An SQL database for persistency of the data:

- Master data
- Transaction logs
- Optimized prices

- Price strategy configuration
- Statistics

4.2 AIR Runtime Services



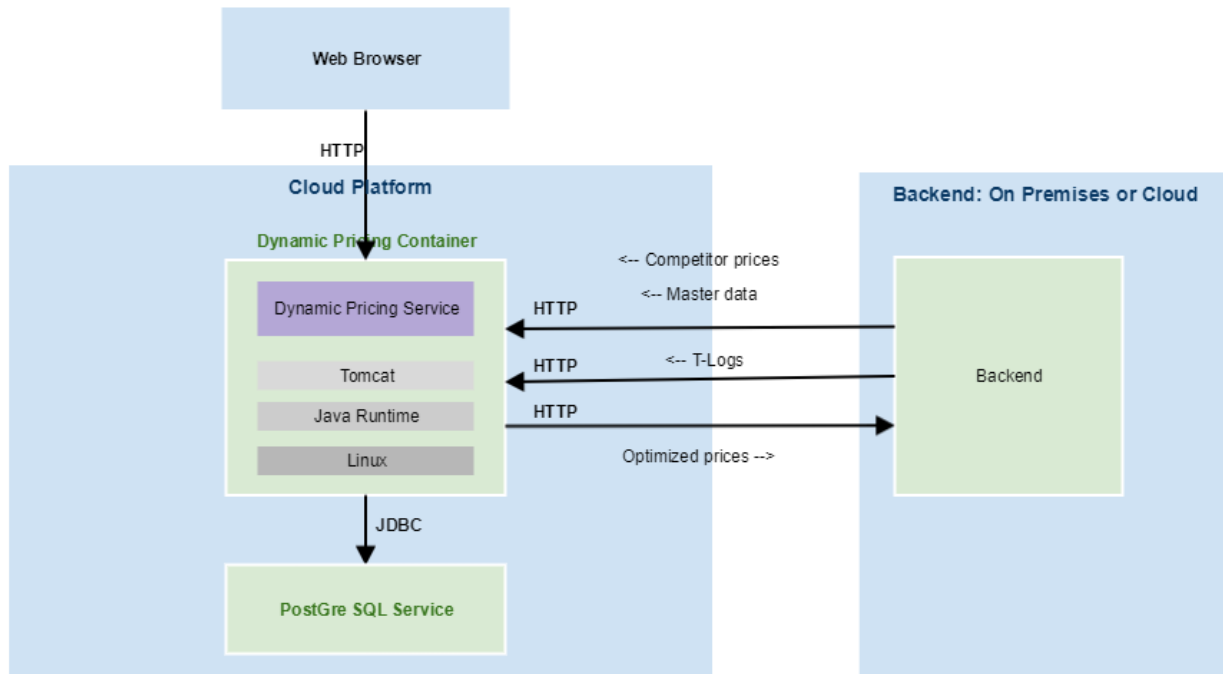
The AIR runtime provides these services to all plugins that it hosts:

- Security: Authentication and authorization
- Interoperability: Communication between plugins of one instance
- Installation, Update, Deployment for plugins
- Portal: A portal frame where plugins can register their user interface front ends.
- Configuration: Wiring of plugins and managing configuration settings
- Messaging: ActiveMQ-based messaging infrastructure
- Scheduling: Quartz scheduling for time-based activation of services

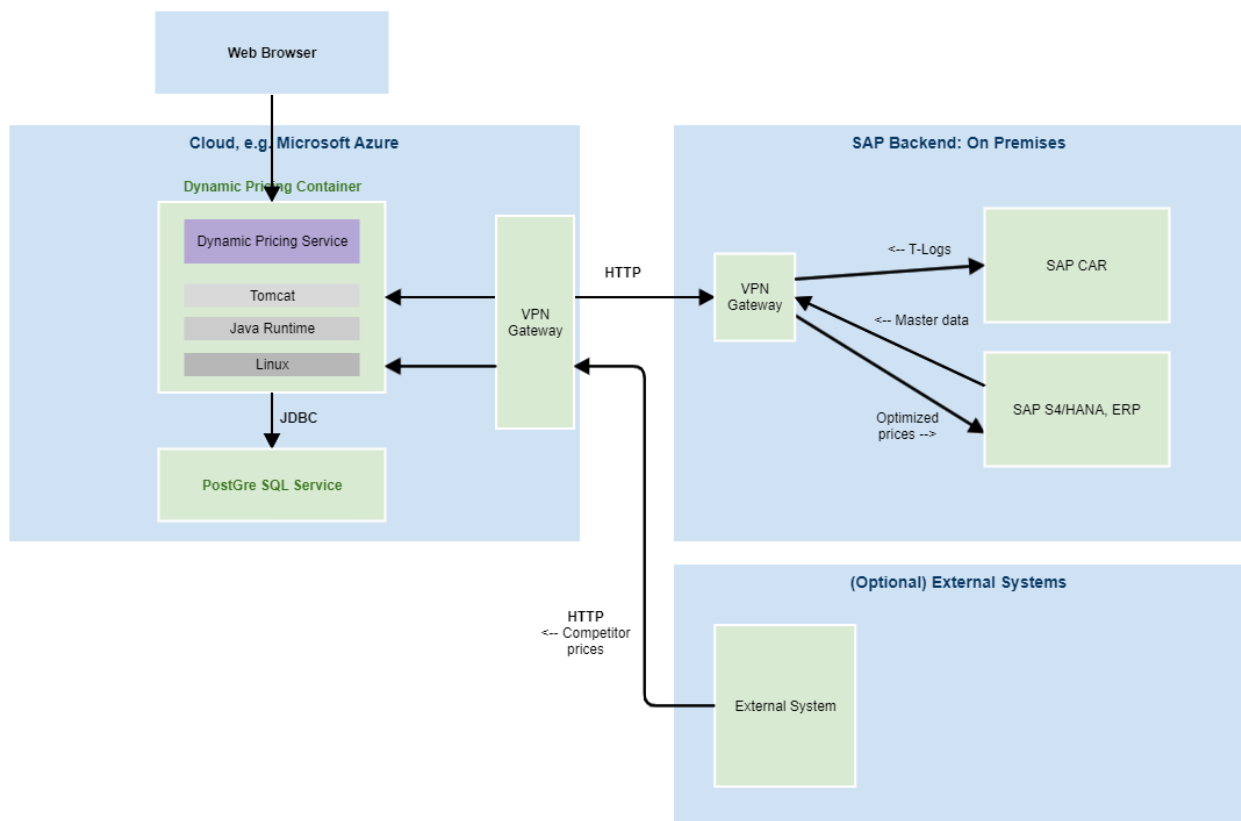
5 Deployment Options

5.1 Cloud Deployment

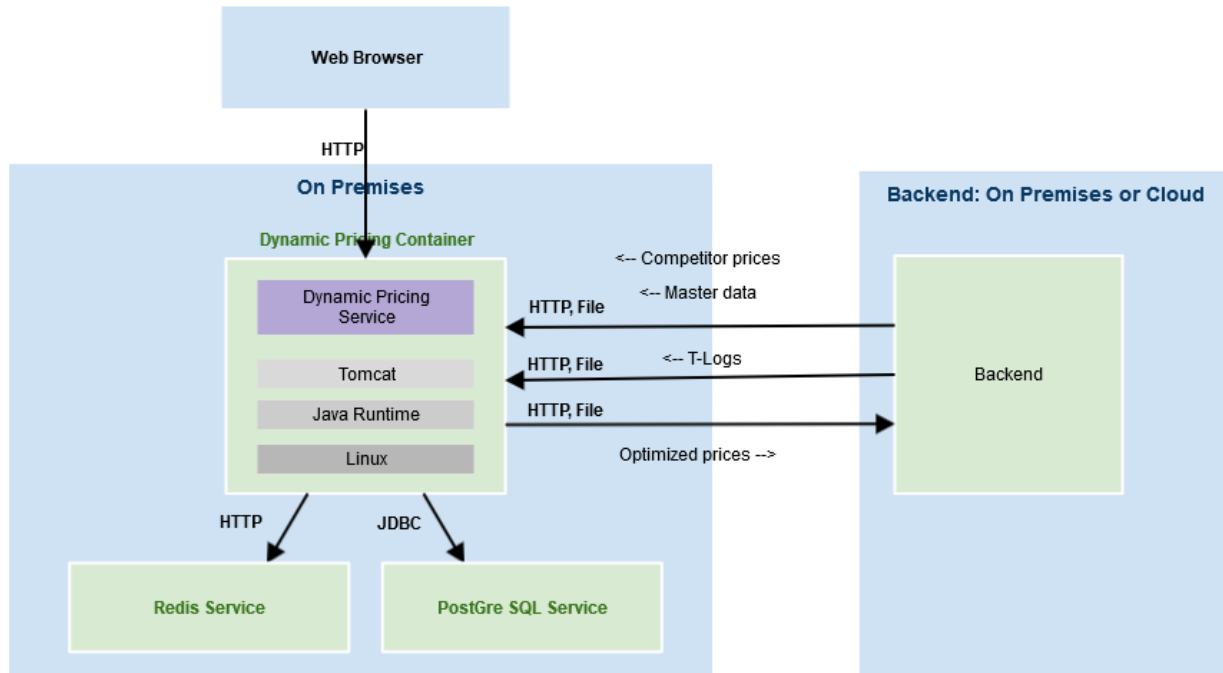
5.1.1 Cloud Deployment with Backend using REST



5.1.2 Cloud Deployment with SAP Backend (on Premises)



5.2 On Premises Deployment



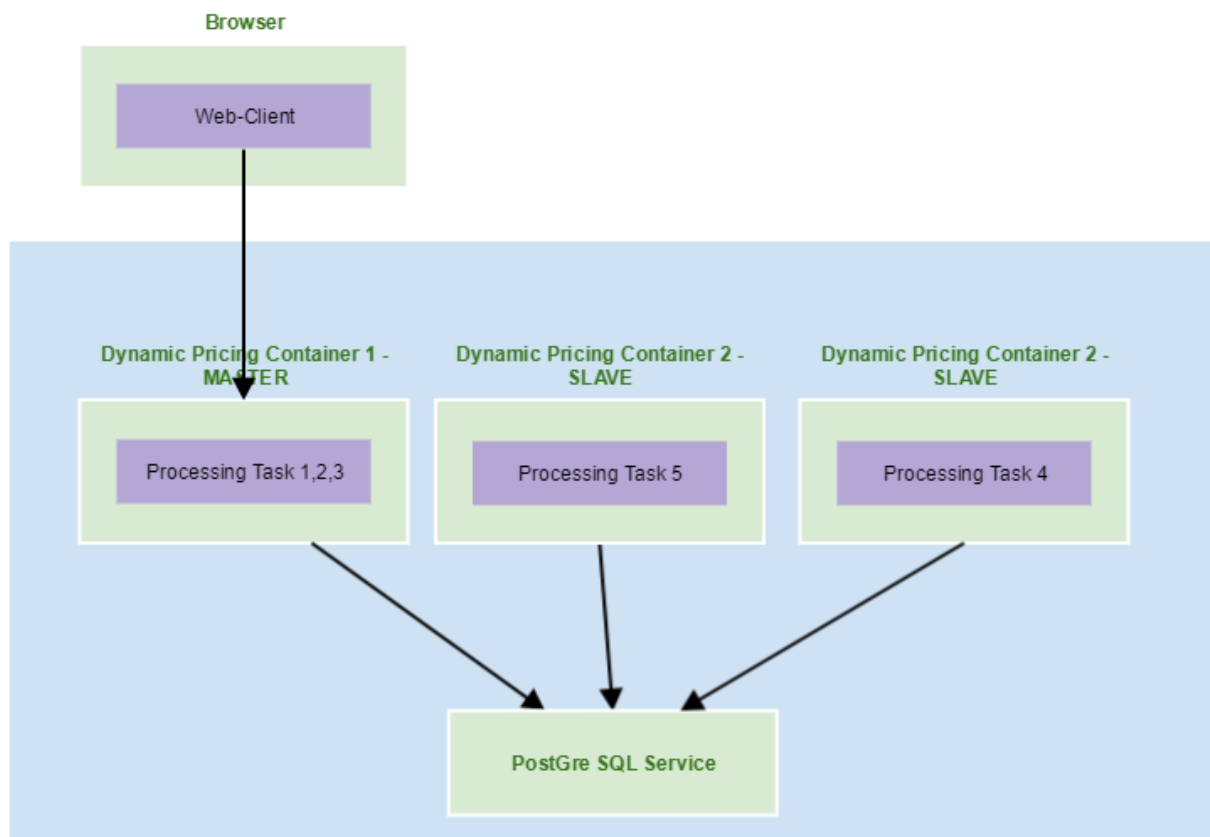
6 Scaling Options

The Dynamic Pricing application supports horizontal scaling both for tenants and business units. See details in the "Sizing Guide".

If you have to run many price computations in parallel, you can scale your containers horizontally.

There is a dedicated master node and one or more slave nodes. The master node provides the Web-Client. You can not login to any of the slave nodes. The slave nodes are configured to process a given amount of tasks for statistics, pricing, import etc.

Each nodes looks up open tasks in the database, then registers for the task and starts processing the task. There is a metric that offers information how many tasks are waiting and how many are in progress. This way an automatic up/down scaling of slave nodes can be implemented.



7 High Availability

7.1 High Availability of Cloud Services

We rely on high available cloud services. We expect high availability of the application runtime services for our containers as well as for the database. E.g. the PostgreSQL database service of the SAP Cloud Platform offers an availability of 99.9%. Internally the service uses more nodes to provide a high availability of the service. The application is prepared to support database node switching and recreates the connection pool if needed.

7.2 High Availability Configuration of the Application

If a container fails, the underlying cloud platform is able to detect a failed application and restarts the application automatically. The user interface then is unavailable for some seconds.

Besides that you can run more application instances besides the master node (see chapter Scaling). If the master node is restarted, the other nodes take over automatically the unprocessed tasks (and also a formerly processed task which status is reverted to unprocessed after some seconds of a node being down). Batch processing tasks are then switched to slave nodes automatically. This feature is used for:

- Processing of import data
- Processing of export
- Processing of statistics
- Processing of price computations

8 Software Development Kit

8.1 Overview

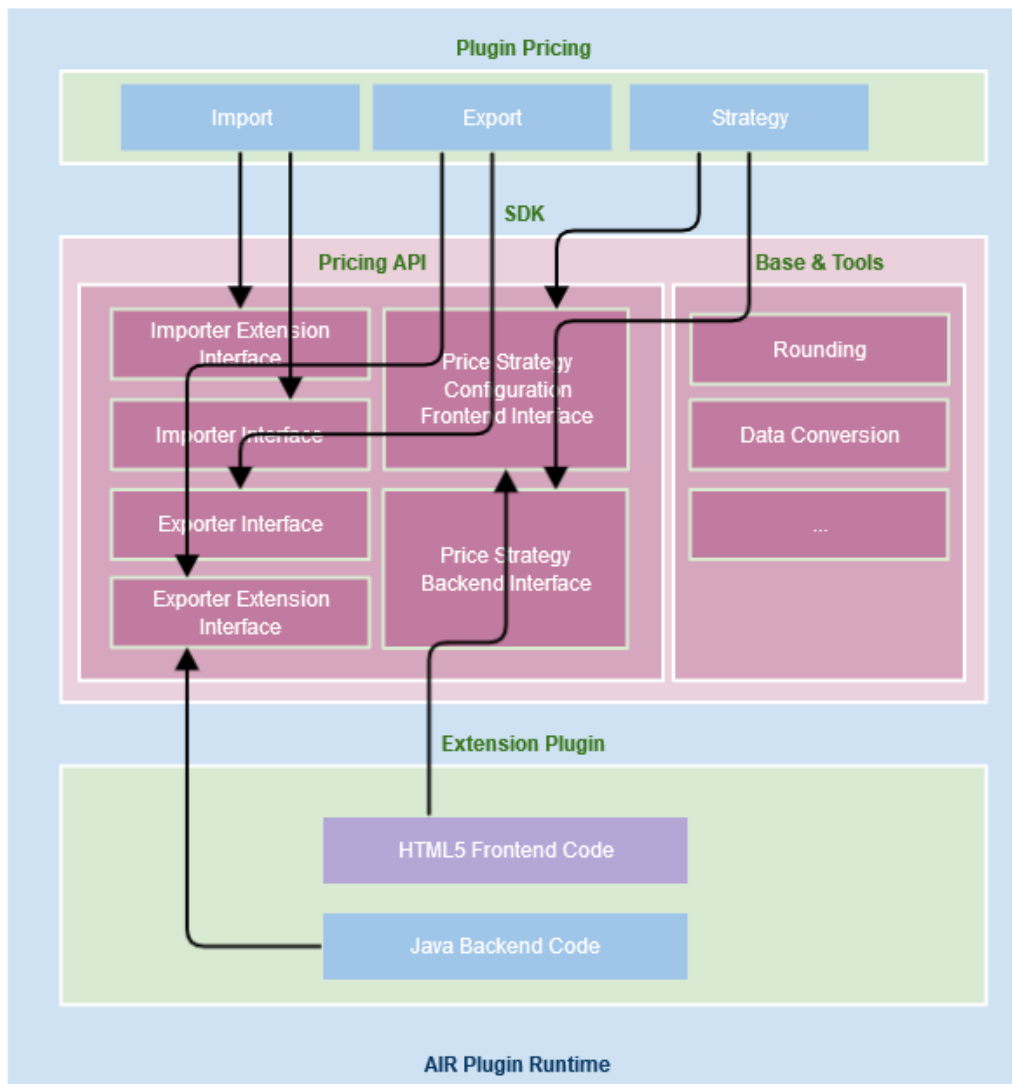
The product comes with a Software Development Kit (SDK) that enables you to:

- Create custom extension plugins
- Extend the standard CSV importers: new data fields, changing behavior of the standard importers
- Extend the standard CSV exporter: new data fields, changing behavior of the standard exporter
- Create new data tables and fields
- Create a new data importer
- Create a new optimized price exporter
- Create a new price strategy including configuration user interface

See more details in the "Extension Guide".

8.2 Integration of Custom Code

Extensions and custom code is packaged as a plugin. It is possible to provide more than one plugin package. The AIR runtime identifies their offered extensions and services (import, export, ...) and registers them automatically so they are available in the pricing application. The extension plugin uses the API of the SDK to implement extensions and custom services. Besides the API, the SDK also offers tools to ease the development of custom extensions.



CONTACT

GK Software SE
Waldstraße 7
08261 Schöneck
Germany

T +49 (0) 3 74 64 84 – 0
F +49 (0) 3 74 64 84 – 15
documentation@gk-software.com
www.gk-software.com