glück kanja

Provisioning-as-a-Service

B2005 Service Description - Commentary

Version: March 2019

The information in this document is subject to change without notice. All statements, information, and recommendations are believed to be accurate but are presented without warranty of any kind, expressed or implied.

In no event shall Glück & Kanja Consulting AG or its suppliers be liable for any indirect, special, consequential, or incidental damages, including, without limitation, lost profits or loss or damage to data arising out of the use or inability to use this document, even if Glück & Kanja Consulting AG or its suppliers have been advised of the possibility of such damages.

Glück & Kanja Consulting AG, Kaiserstraße 39, D-63065 Offenbach am Main, Germany
HRB 12381 Offenbach am Main, Steuer-Nr.: 035 234 46203
Vorstände: Michael Breither, Christoph Fausak, Harald Glück, Christian Kanja, Felix Storm
Aufsichtsratsvorsitzende: Lilian Glück

Table of Contents

Abstract

The Glück & Kanja Software Provisioning Service delivers a standardized package lifecycle process for RealmJoin Software packages. With the Software Provisioning Service, all software packages are managed by a dedicated team of experienced engineers providing all services from initial request to decommisioning of software packages.

Roles

Application Owner

The Application Owner role is a customer employee that is the business owner of the application and is responsible for the business delivery, functioning and services of the application. The Application Owner is also the custodian of the data in the application. The Application Owner is requesting new releases from the Customer Delivery Team.

GK Customer Delivery Team

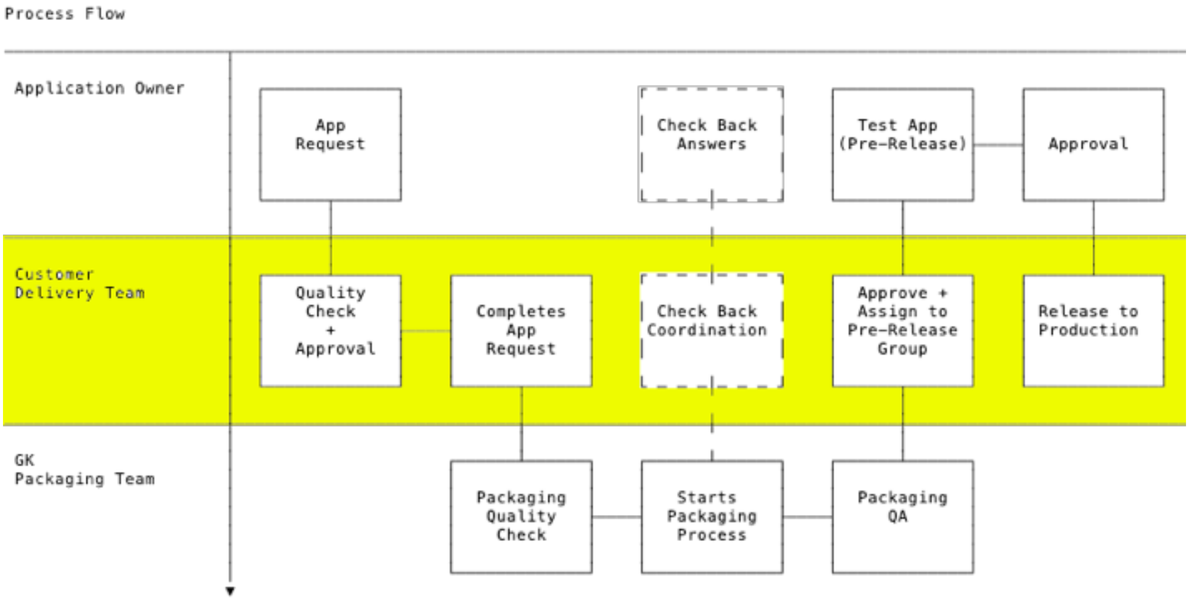
The Customer Delivery Team is the organization, administration, and supervision of the processes to get the applications requested by the Application Owner delivered. The team collects the business and technical functions needed to successfully achieve what a client expects to receive. An important task is to verify the quality of the request and prepare the decision of the package type to be created. When ready the Customer Delivery Team hand over the requirements and the setup binaries of the requested application to the Packaging Team.

GK Packaging Team

The GK Packaging Team develops the requested setup packages with a standardized and partly automated process to create high quality deliverables. The GK Packaging Team is verifying the overall quality and package type decision. The team checks back to the Customer Delivery Team if the uploaded binaries are incomplete or other questions arise from the sources. An automated and manual QA-check is done by the GK Packaging Team before delivering the package back to the Customer Delivery Team.

Overall Process Flow

The overall process is visualized in the below process flow. The yellow segment describes the responsibility of the Customer Delivery Team during a package creation.



To realize an efficient and auditable process the communication between the Application Owner and the Customer Delivery Team shall not use direct email. Instead the initial request needs to be made on a request portal described later in this document. By using the portal the identity of the application owner is proofed and additional information (cost center, authorized representative, sponsor, etc) can be collected.

When the request is created the ongoing communication is ticket driven within the Customer Delivery Team but the Application Owner will be able to stay in the loop by using standad email but connected to the provisioning ticket.

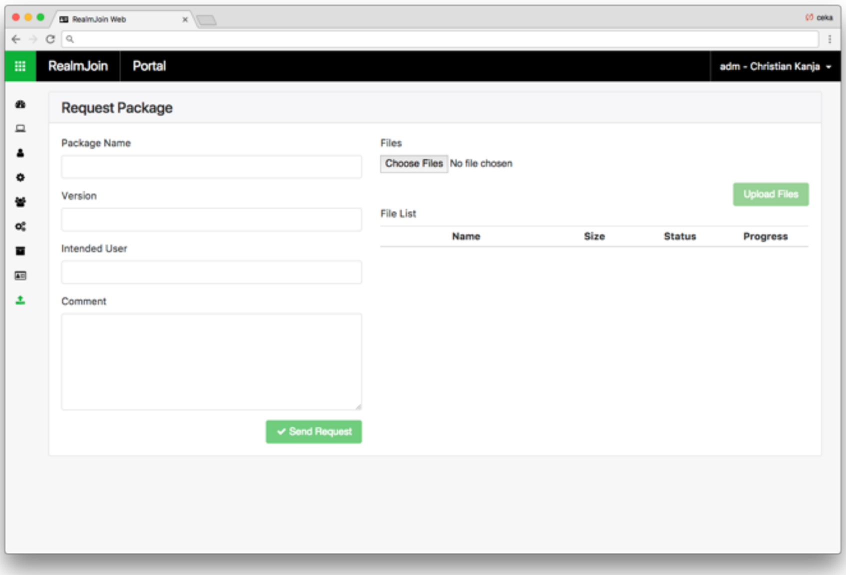
In addition to the above flow the Customer Delivery Team can provide the service to create AzureAD groups for preview and production use of the requested package and bind these groups to the parameterized package deployment assignment in RealmJoin.

Similar tasks are related to the complete lifecycle of an application including the maintenace of updates, dependencies, decommissioning and archiving.

The Package Lifecycle

Provisioning Portal

Every creation or updating request for deployment packages should be made in the AzureAD integrated provisioning portal provided by GK. A first preview of this portal is shown below but the implementation will be extended to provide PRF blocks (secure uploading capabilities are already available but some data should be extraced and used for the flow) and additional metadata with some to be discussed with the customer to provide optimized integration in billing and controlling flows. Aalready known metadata is the cost center code.



Request Package

Package Name

Version

Intended User

Comment

Files

No file chosen

File List

Name	Size	Status	Progress

Quality Check Procedures

The QA process includes a logical and technical part. First of all a logical QA with an investigation about already existing packages of the same or similar content is started. This includes a check for newer versions which in general should always outvote a request for an older version. In addition a basic background check about the vendor and the trustworthyness of the software is done.

If anything fails in the logical QA a decision tree settled with the customer is started. The regular outcome is the use of the existing package or the cancellation of the request. If the application owner insists against the logical QA an escalation to a customer escalation named person may start and the decision may be made to continue against the logical QA status.

When the logical QA is done a technical QA starts and the compatibility with the modern workplace is checked. This includes operating system compatibility, requirements of legacy technogies (fileserver, domains, etc), outdated technologies (old browsers, etc), dependencies (.net, Java, other). In addition the provided setup package is checked for completeness and functionality.

Again, If anything fails in the technical QA a decision tree settled with the customer is started.

Creation of the package

Based on the provided binaries and documentation a package format and parameterization strategy is decided. This starts the technical creation process with the GK packaging team (Service Catalogue B2002) and coordinates everything including the joined automated and manual test process to approve the package for pre-release. Depending on the type of package request and the requirements of the application owner the application owner is involved in the test process.

Management of prerelease packages

In general a Provisioning-as-a-Service-Contract includes the handling of Azure AD application groups. If this is the case, the Customer Delivery Team maintains the application groups during the lifecycle of a package. When a pre-release package is avaiable a respective group is created and the pre-release package is assigned.

The assignment of users should not be part of the deliverables. This decision should be done by people responsible for the usage, licensing and target group of a package.

Release to production

If a package is approved for release the Customer Delivery Team a respective group is created and the released package is assigned. Depending on several parameters the deployment details are varying: mandatory vs optional, staggered deployment, background deployment, etc.

Lifecycle and evergreen approach

When a package is released and in production the Customer Delivery Team works with the application owner to set an update stragy in place. If it’s a well known software the Customer Delivery Team can add the package to the list of managed packages and trigger the autonomous release of application updates when available. The same process is done for dependencies like runtimes, frameworks or browsers.

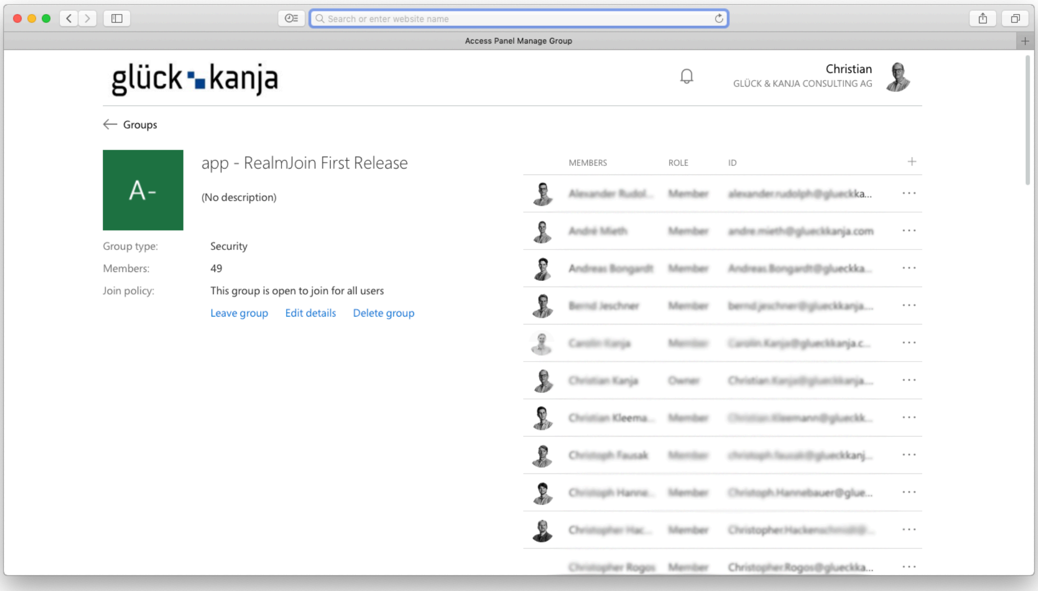
Decommissioning and Archiving of Packages

If a package is end-of-life by superseeding or not used anymore the decomissioning is done by the Customer Delivery Team if requested by the application owner or triggered by market knowledge. Uninstall packages are created if necessary and the AzureAD groups and assignments are outphased. The package is still available archived within the provisionong backend.

Group and User Management

In general a Provisioning-as-a-Service-Contract includes the handling of Azure AD application groups. The assignment of users should not be part of the deliverables. This decision should be done by people responsible for the usage, licensing and target group of a package.

Azure AD provides a web based management of application groups which can be used without customization:



Search or enter website name
Access Panel Manage Group
glück kanja
Christian
GLÜCK & KANJA CONSULTING AG
Groups
A-
Leave group
Edit details
Delete group

app - RealmJoin First Release

(No description)

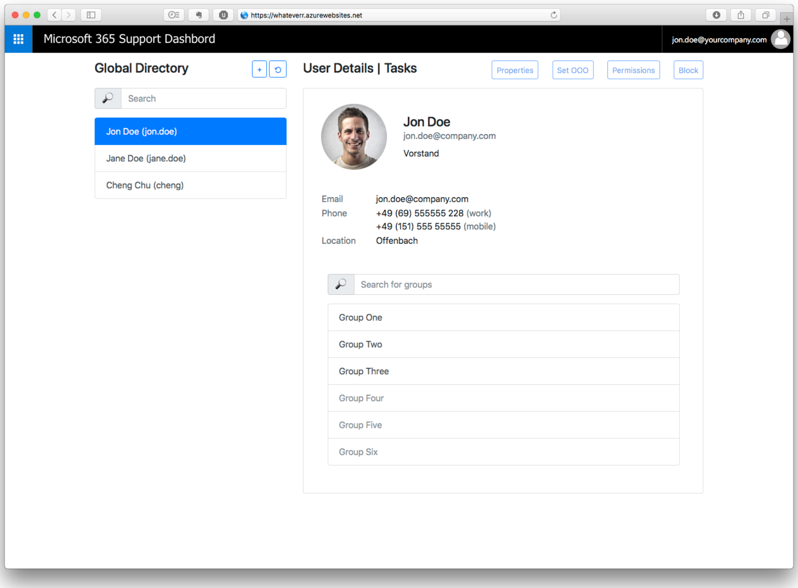
Group type: Security

Members: 49

Join policy: This group is open to join for all users

MEMBERS	ROLE	ID
[redacted]	[redacted]	[redacted]
[redacted]	[redacted]	[redacted]
[redacted]	[redacted]	[redacted]
[redacted]	[redacted]	[redacted]
[redacted]	[redacted]	[redacted]
[redacted]	[redacted]	[redacted]
[redacted]	[redacted]	[redacted]
[redacted]	[redacted]	[redacted]
[redacted]	[redacted]	[redacted]
[redacted]	[redacted]	[redacted]
[redacted]	[redacted]	[redacted]

GK could provide a web based portal to allow other dedicated teams to manage all or some application groups without being part of the overal AzureAD management.



Jon Doe (jon.doe)

Jane Doe (jane.doe)

Cheng Chu (cheng)

Jon Doe
jon.doe@company.com
Vorstand

Email jon.doe@company.com
Phone +49 (69) 555555 228 (work)
+49 (151) 555 55555 (mobile)
Location Offenbach

Group One

Group Two

Group Three

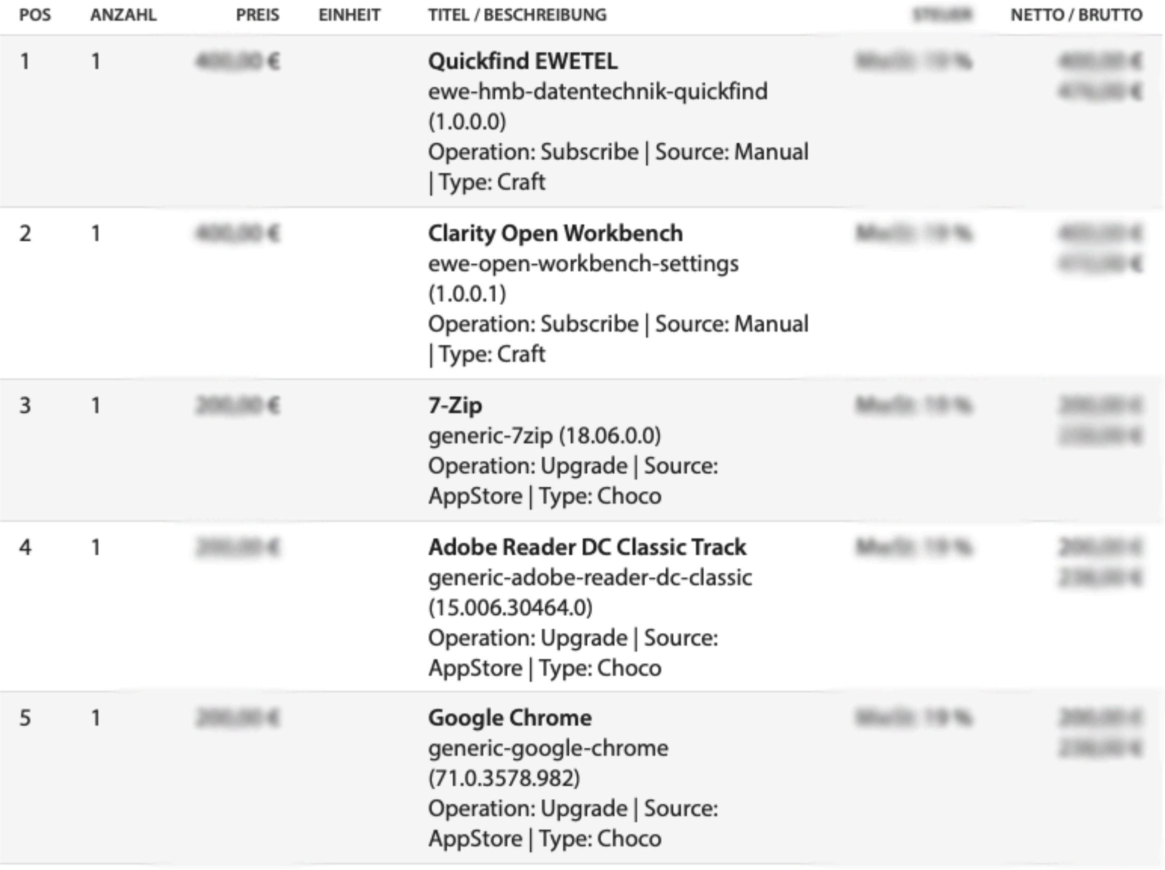
Group Four

Group Five

Group Six

Billing

Based on the metadata of the provisioning request GK can offer to provide detailed lists with all details about the provisioning lifecycle on a monthly base:



POS	ANZAHL	PREIS	EINHEIT	TITEL / BESCHREIBUNG	[redacted]	NETTO / BRUTTO
1	1	[redacted]		**Quickfind EWETEL** ewe-hmb-datentechnik-quickfind (1.0.0.0) Operation: Subscribe | Source: Manual | Type: Craft	[redacted]	[redacted]
2	1	[redacted]		**Clarity Open Workbench** ewe-open-workbench-settings (1.0.0.1) Operation: Subscribe | Source: Manual | Type: Craft	[redacted]	[redacted]
3	1	[redacted]		**7-Zip** generic-7zip (18.06.0.0) Operation: Upgrade | Source: AppStore | Type: Choco	[redacted]	[redacted]
4	1	[redacted]		**Adobe Reader DC Classic Track** generic-adobe-reader-dc-classic (15.006.30464.0) Operation: Upgrade | Source: AppStore | Type: Choco	[redacted]	[redacted]
5	1	[redacted]		**Google Chrome** generic-google-chrome (71.0.3578.982) Operation: Upgrade | Source: AppStore | Type: Choco	[redacted]	[redacted]

Beside the application name, requester name, cost center, dates, etc. it may also be of interest to controlling to include zero-cost items showing cancelled provisionings and decomissioned packages.