Nice Template! Can We Buy It?

The Reality of Turning a SaaS solution for German Utilities into a Product

Christopher Graw, rku.it GmbH

rku IT

rku-it.de

Agenda

rku IT

1 **Evolution of NextGen**

2 Template Vendition

3 Demonstration

rku.it at a Glance

Established IT Service Provider (since 1961)

Delivering stable, scalable, and future-oriented digital solutions across Germany.

Company Snapshot

2 locations
Herne (Headquarter) Leipzig (Branch Office)

470+ employees
15 shareholders
130+ customers
€100M+ revenue

Mission

Providing innovative and tailored IT solutions.

Enabling digital public services and sustainable, smart environments.

User & Device Landscape

15,000+ users
6,500+ managed endpoints

Industry Expertise

Deep domain knowledge in **Utilities, Mobility, and Public Sector**

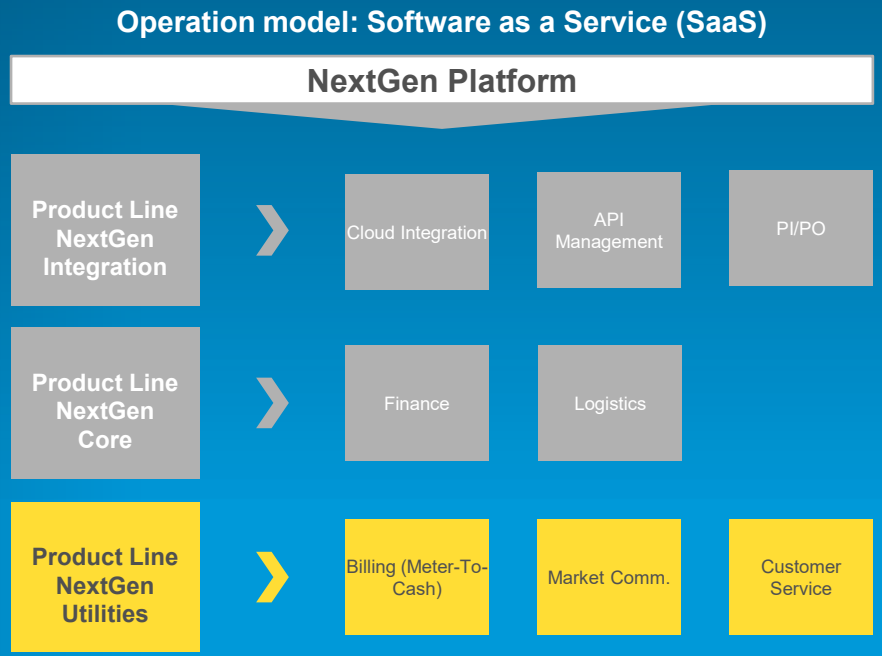
Operational Scale

~5 million metering points managed for energy providers
80+ production systems (ERP environments)
~1,400 Windows servers
40+ municipal utilities served

NextGen at a glance

Platform solution for German Utilities and Public Transport

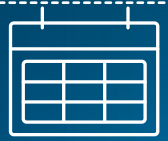
- NextGen is a collection of three interconnected SaaS solutions for ERP, Utilities Billing and Integration, based on SAP S/4HANA and BTP.
- NextGen Utilities is our Template solution for German Utilities Billing based on SAP S/4HANA Utilities.
- NextGen Integration establishes the SAP S/4HANA systems as the centerpieces of the entire system landscape.



Evolution of NextGen

Timeline

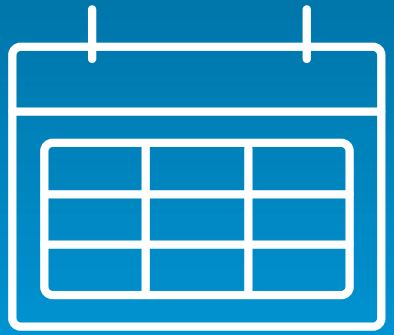
2013 - 2020	2018 - 2019	2020 - 2022	2022 - 2024	2024 - 2027	2025 - 2027
Harmonized Development in SAP IS-U	Conception of NextGen as a SaaS solution	Product Setup NextGen Core and NextGen Integration	Product Setup NextGen Utilities Transformation projects for NextGen Core	Transformation projects for NextGen Utilities	Focus on Template Vendition



Evolution of NextGen

(I) Harmonized Development

- The Harmonization Project started with the introduction of the Central Development System in 2013, Central Maintenance of Customizing was added in 2017.
- The first major success of this approach was the simultaneous introduction of IDXGC* for all our Utilities customers in 2015.

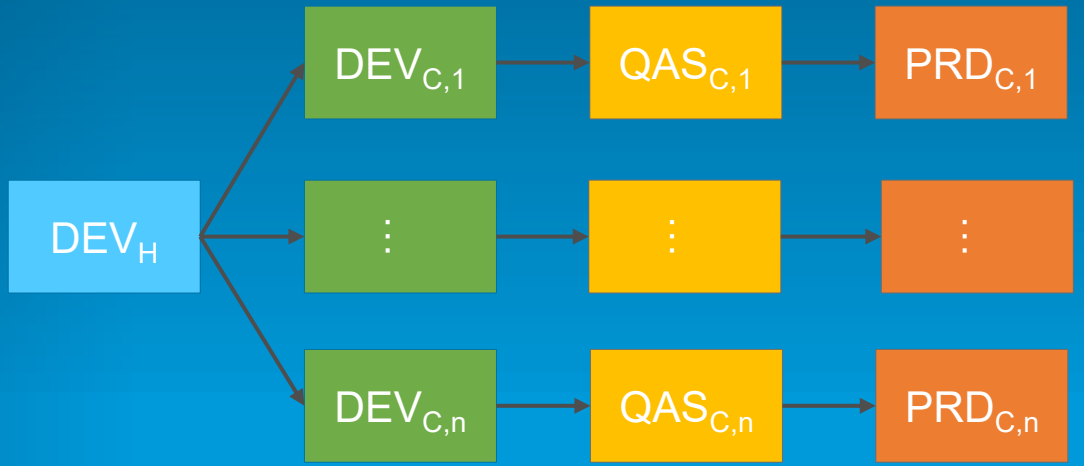


* Intercompany Data Exchange for German Utilities – Common Layer

Evolution of NextGen

(I) Harmonized Development

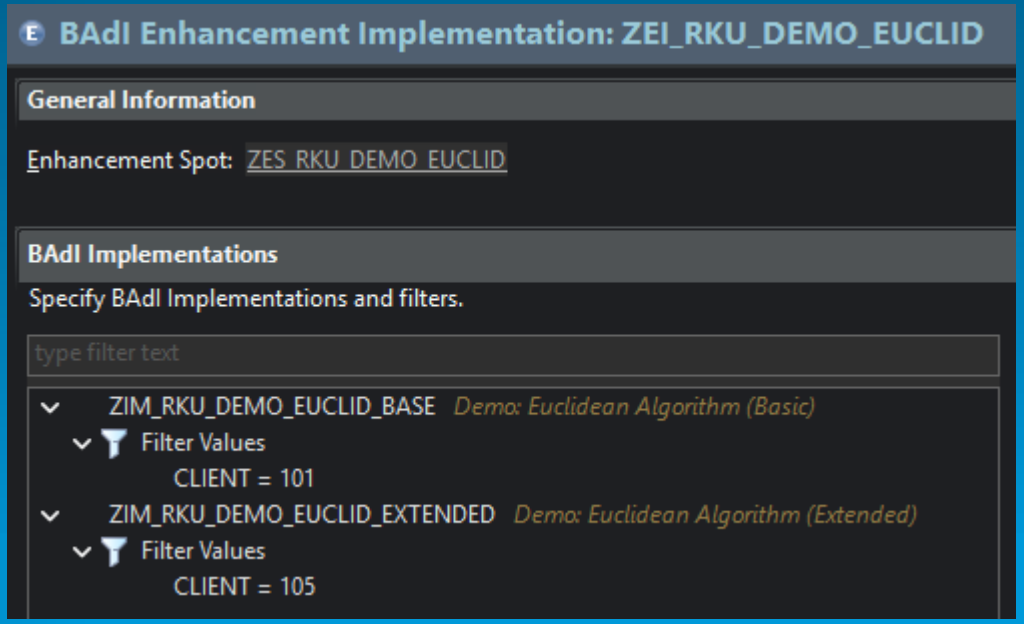
- The Central Development system is used for all Development objects, that are supposed to be equal across all customer systems.
- This covers many objects underlying regulatory requirements, most notably Intercompany Data Exchange
- Objects closely linked to individual customers' corporate identity and design, e.g. products or forms, are usually exempt



Evolution of NextGen

(I) Harmonized Development

- The Harmonized Development allows customer-specific extensions at certain extension points.
- The extensions are realized using custom BAdIs and client filters.
- One Implementation can be (and often is) used for several clients.
- The usage of client filters also allows the implementations to be developed on the central system (DEV_H) itself.



Evolution of NextGen

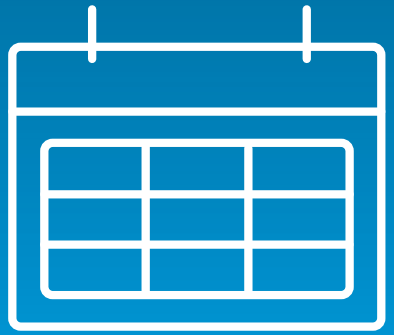
(I) Harmonized Development

Pros	Cons
+ The used technology (BAdI with filters) is 100% SAP standard	– Decentralized maintenance of implementations (each BAdI has its own ABAP interface and Enhancement Spot)
+ Additional filters (e.g. company code) can be used as needed	– The maintenance of BAdI Implementations and filters is generally cumbersome.
	– Focus on singular extension points, hard to extend functions or processes on a large scale
	– Not suited to replace a function entirely by a different implementation

Evolution of NextGen

(II) Action Framework

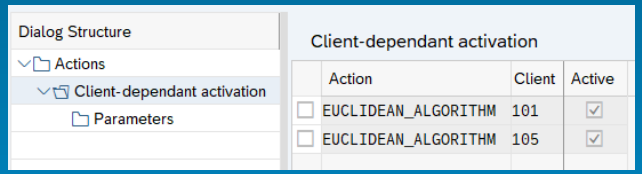
- The Action Framework was developed for a large customer of ours (Many subdivisions, several clients operated in a SaaS model, >10 clients and ~90 company codes in total) in 2018
- It allowed the execution of specific functions depending on the current client and company code
- This approach is essentially a simplified version of the Switch Framework and Business Functions of the SAP standard

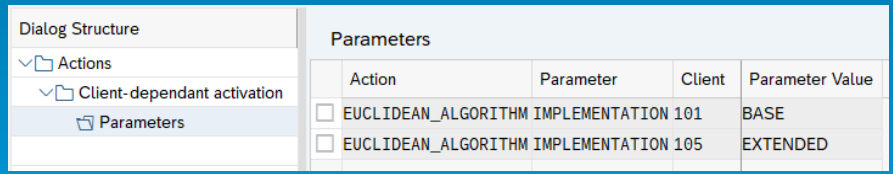


Evolution of NextGen

(II) Action Framework

- Actions are defined in a client-independent customizing table.
- Activation is managed on the level of clients and company codes*.
- Actions can define Parameters with distinct values for each client and company code.
- The Status of an Action and Parameter values are accessible at runtime by a dedicated ABAP class





* removed in the example and demo respectively

Evolution of NextGen

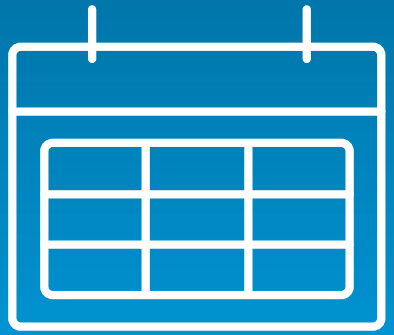
(II) Action Framework

Pros	Cons
+ Light-weight implementation	– Focus on singular extension points, hard to extend functions or processes on a large scale
+ Centralized maintenance of extensions	– Not suited to replace a function entirely by a different implementation
+ Close to SAP Standard (Custom Table with Maintenance View)	

Evolution of NextGen

(III) Conception as a platform solution

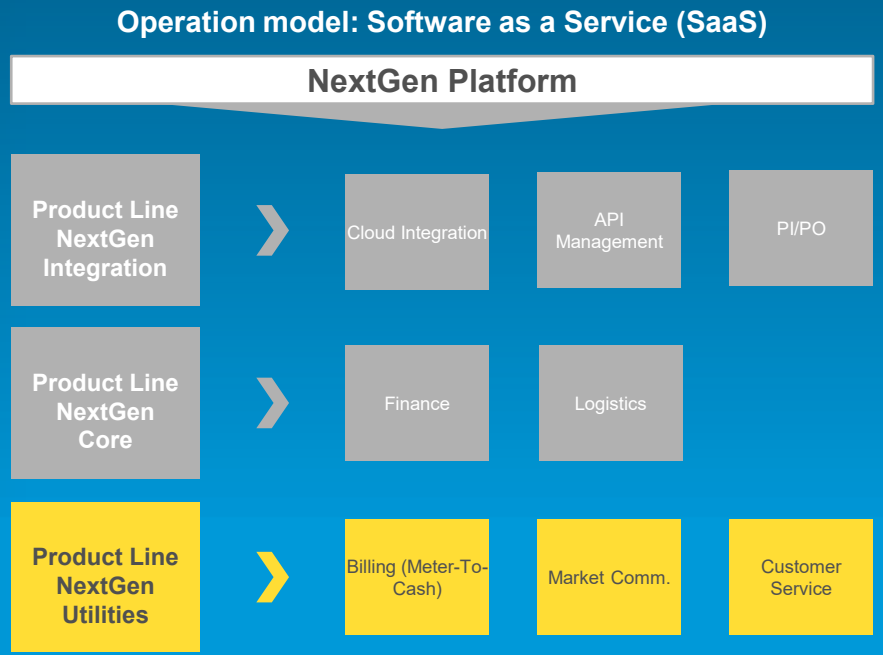
- When SAP announced the end of life for SAP R/3, rku.it considered several alternatives to approach the transformation to S/4HANA.
- The result was to develop NextGen, our platform solution with three distinct product lines (Core, Utilities and Integration)
- The initial concept for NextGen was written in 2019 and presented internally in January of 2020.



Evolution of NextGen

(III) Conception as a platform solution

- NextGen Core and NextGen Utilities serve as the platform's backbone, providing a stable core for the customer-specific implementation.
- NextGen Integration provides connectivity to the customer's system landscape, including On-Premises Integration as well as Cloud Integration.
- NextGen in a Nutshell: Stable Core, Flexible Environment!



Evolution of NextGen

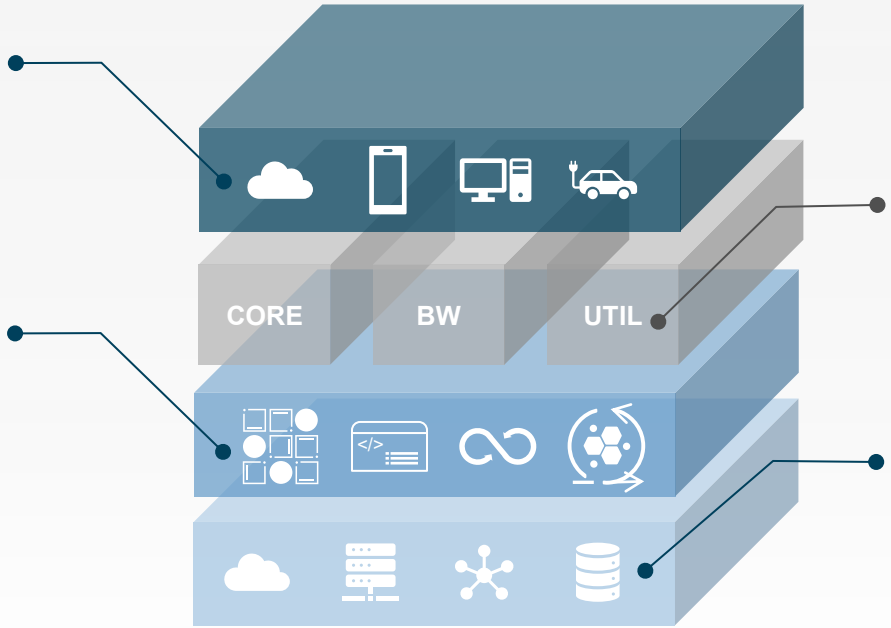
(III) Conception as a platform solution

Integration Layer

Integration strategy, Process and Data Integration, API Management

DevLayer

Service Architecture, Release Management, Quality Assurance, DevOps, Software Development Lifecycle



Business Layer

Business Processes (Basis / Premium Services) for CORE, BW and UTILITIES

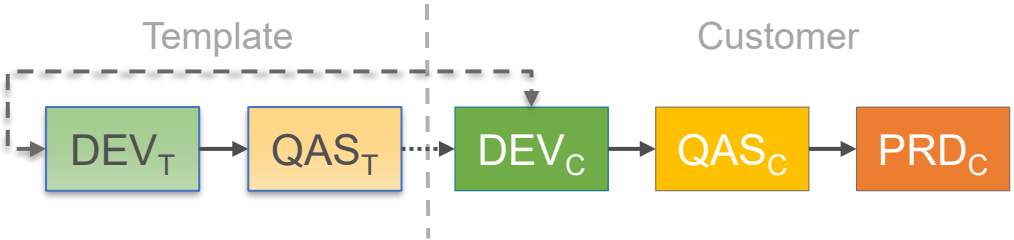
Basis Layer

Database and Application Administration, Network, Cloud Configuration, Identity and Access Management

Evolution of NextGen

(IV) System Landscape

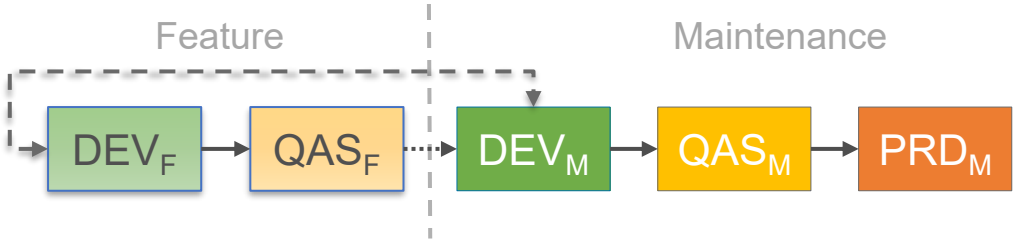
- In the first model, there are dedicated Development and Test systems for the Template.
- New Template features and changes are developed and tested in a „Lab Environment“ and shipped to the customer systems after successful internal testing.
- Customer-specific development, installation of add-ons, and Customizing are only carried out in the Customer Development system (DEV_C)
- All features and changes are tested in the Customer QA system and deployed to production by means of regular Releases.



Evolution of NextGen

(IV) System Landscape

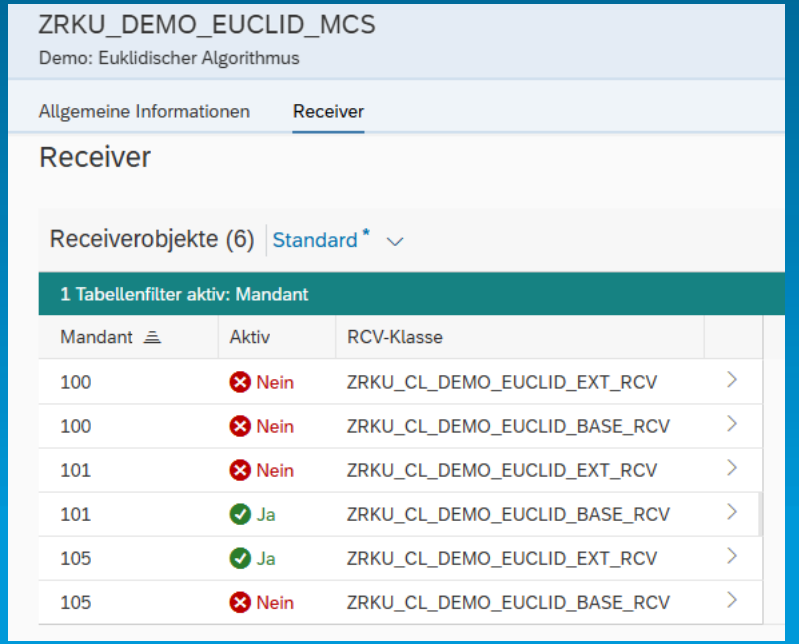
- In the second model there are separate development systems for long-term (Releases, projects) and short-term (Hotfixes) development.
- Customer-specific development, installation of add-ons, and Customizing are carried out in the Feature Development system (DEV_F) and are available in all systems.
- All new features and changes are tested in the Feature QA system.
- Outside of Major Releases, the Maintenance QA system is only used to test Hotfixes. To prepare Major Releases, it is also used for Acceptance tests on a larger scale.



Evolution of NextGen

(V) Service Catalog

- All Custom Code is organized in services. Each Service has its own ABAP package and Interface.
- A service can have several implementations. Implementations can be “Basis” or “Premium”.
- Implementations can be activated or deactivated individually for each client.
- The Services and their activation status are maintained in a central Service Catalog.



Evolution of NextGen

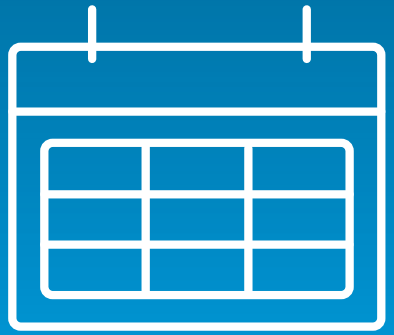
(V) Service Catalog

Pros	Cons
+ Ability to organize the entirety of the custom code base	– All custom code depends on the Service Catalog (Single Point of Failure)
+ Centralized maintenance of implementations	– The Service Catalog itself is complex and heavy in maintenance
+ Dependencies between services can be analyzed and checked	
+ The notion of a service is directly linked to versions control (Main Package <> gCTS repository)	

Evolution of NextGen

(VI) Product Development and Transformation

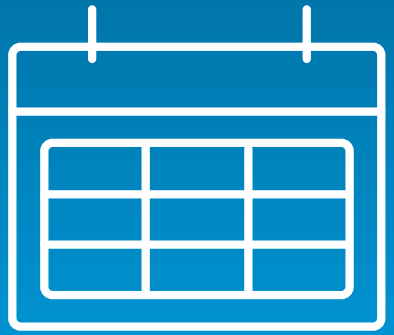
- The three product lines were developed in internal projects starting in 2020 (Core, Integration) and 2022 (Utilities), respectively.
- The first transformation projects started in 2021 (Core) and 2024 (Utilities), respectively.
- While the transformation is mostly complete in NextGen Core, 10 transformation projects in NextGen Utilities are yet scheduled to complete until the end of 2027.



Evolution of NextGen

(VII) Product Vendition

- During the first transformation project in 2024, we got the first inquiries, if we were willing to sell our template to customer, who wanted to host the system themselves.
- Throughout the rest of 2024 we discussed the two relevant questions:
  - Do we want to do it? (Strategic decision) – “Yes”
  - Can we do it? (Technical / Organizational / Legal) – “Let’s see …”



Agenda

rku IT

1 Evolution of NextGen

2 Product Vendition

3 Demonstration

Product Vendition

(I) Introductory questions

“What exactly is our Template?”

“It is a (partially) preconfigured S/4HANA Utilities System (Customizing, Custom Code, Documentation), that is used as the basis for building our customers’ systems.”

Product Vendition

(I) Introductory questions

rku IT

“What do we deliver?”

- NO SAP Code!
- All Custom Code that belongs to the template implementation
- All Customizing from the “Golden Client” (101)
- Customizing from a sample implementation or demo client
- A standard set of Fiori catalogs and associated roles
- Some additional content (e.g. UPIL products)
- Documentation (Installation Instructions, Release Notes, User Manuals, …)

Product Vendition

(I) Introductory questions

rku IT

“How do we deliver?”

- Initial Shipment by means of an external transport request (Workbench) and client export (Customizing)
- Regular Updates by external transports according to the Release Schedule
- Transport files and supplementary files like documentation are provided on a secure file server hosted by rku.it

- Alternatively, we considered the option of providing access to the Template via a large gCTS monorepository that is updated according to the Release Schedule

Product Vendition

(II) Prerequisites

In order for the Template to be shipped to a customer system, it is necessary to make sure that the target system is close enough to our source system. This comparison must cover at least the following topics:

- Installed SAP Standard Modules and Add-Ons and their Product / Component versions
- Installed SAP Notes and their respective correction instructions
- Active Business Functions
- Installed third party solutions and their product versions (list of external transports)

Product Vendition

(II) Prerequisites

A successful delivery of the Template requires the existence of concrete boundaries and the ability to differentiate between the Core or Template and customer-specific implementation. This includes the following:

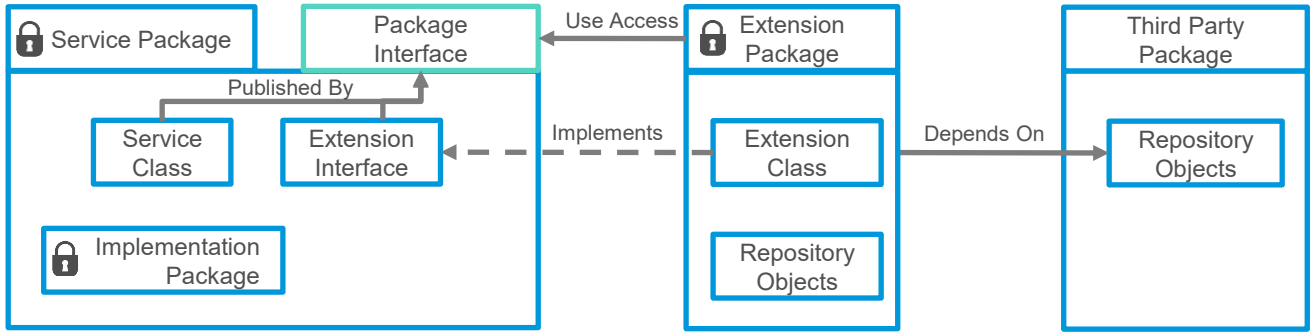
- Workbench:
  - Separate Package Hierarchies and Software Components for Template content, customer-specific extensions, and third-party extensions
  - Usage of a dedicated SAP namespace for the Template content (not strictly necessary, but highly recommended)

- Customizing:
  - Classification (Template, Customer, or Shared with separate naming conventions) on the level of individual IMG activities, View Clusters, Views and Tables

Product Vendition

(II) Prerequisites

- All external dependencies (customer-specific or third-party) of the Template need to be encapsulated to ensure the Template can be delivered without syntax errors.
- This is done by using an additional layer of Extension or Proxy Packages.
- Only objects in these packages may directly call customer-specific or third-party logic.
- To integrate the extensions into the template, the Service Catalog is used in combination with custom BAdIs.



Product Vendition

(II) Prerequisites

In addition to the technical setup and concise definition of the Template, a few other things need to be clarified to ensure a successful delivery:

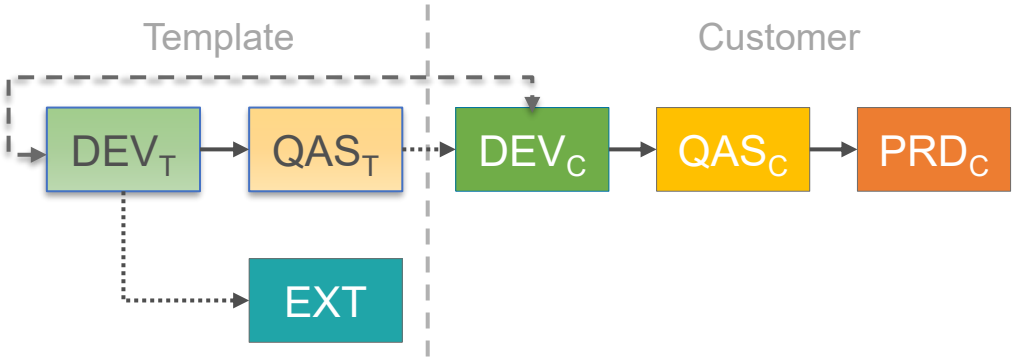
- Documentation:
  - Installation Instructions (Components, Business Functions, order of transports, necessary IMG activities, etc.)
  - Technical Design Documents
  - User Manuals

- Legal Matters:
  - Coding may only be shipped, if the vendor actually holds the right to redistribute it. This needs to be checked for every piece of coding that ends up in the delivery.
  - If the Template relies on third-party solutions to function, the buyer needs to be informed of the products and licenses they need to acquire as well.

Product Vendition

(III) System Landscape

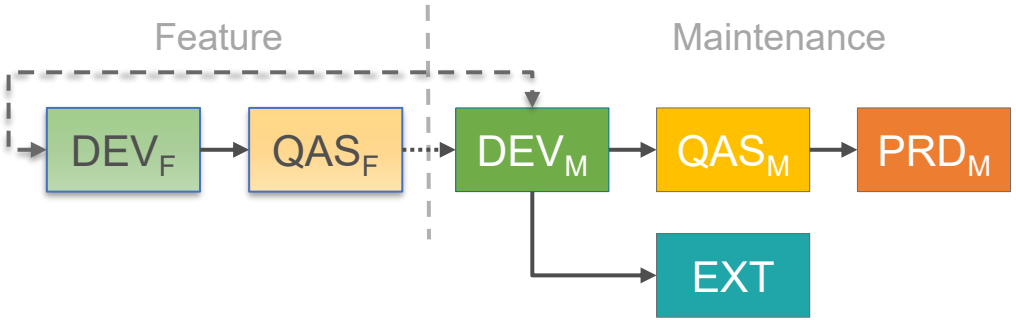
- A new system for external delivery (EXT) is added to the system landscape.
- It receives all regular Release transports, and Hotfixes once they are retrofitted to DEV_T.
- In this scenario, EXT always contains a stable release. If new features are to be delivered, transports need to be forwarded to EXT manually.
- If delivery is done via gCTS, this is likewise possible by deploying Feature branches to EXT and exposing them.



Product Vendition

(III) System Landscape

- A new system for external delivery (EXT) is added to the system landscape.
- It receives all regular Release transports and Hotfixes from the main systems.
- When creating transports for external releases in EXT, filters can be used to include or exclude untested Hotfixes.
- If delivery is done via gCTS, this distinction is likewise possible by exposing the main branch as well as Hotfix branches.



Product Vendition

(IV) Organizational Requirements

- Release Management
  (Stable Releases for Shipment/Delivery)
- Product Office
  (Ownership for Core/Template)
- Service Management
  (Ownership for Processes and Documentation)
- Quality Assurance
  (Code Guidelines, Legal Status, Documentation)

Agenda

rku IT

1 Evolution of NextGen

2 Product Vendition

3 Demonstration

Demonstration

Scenario

- We use the gCTS BAdI provided by SAP (GitHub).
- Given that it is Open Source (Apache 2.0), we may redistribute it as part of our delivery.
- We modified the SAP logic by adding a callback to our external Transport Management System (RevTrac), which is an external dependency.
- To assure an error-free delivery, this extension is encapsulated in a BAdI.
- Objects in the extension package can be filtered from delivery transports.

Disclaimer

The reproduction, duplication, distribution, and/or modification of any content and representations contained in this publication, as well as any use thereof – regardless of purpose or form – is permitted only with the express written consent of rku.it GmbH. rku.it or other products and services mentioned in this document, as well as their respective logos, are registered trademarks. German trademark law applies.

The information provided in this document has been prepared by rku.it GmbH solely for informational purposes. While all reasonable care has been taken in its preparation, rku.it GmbH assumes no liability for the accuracy, timeliness, or completeness of the content and representations contained herein.

Content referenced via external links is not provided by rku.it GmbH and serves solely as supplementary information. rku.it GmbH assumes no responsibility for the accuracy, timeliness, or completeness of such external information.

Any strategies, future developments, products, services, and/or platforms described in this publication or related presentations may be changed or discontinued by rku.it GmbH at any time without prior notice and without stating reasons.

The information contained in this publication does not constitute a guarantee or commitment and does not establish any legal obligation to deliver products or services. Forward-looking statements are subject to various risks and uncertainties that may cause actual results to differ materially from expectations. Such statements reflect the view at the time they were made.

Contact

Christopher Graw

Solution Architect

christopher.graw@rku-it.de