frends

Integration Delivery Guide 2023

Version 6.3

Copyright Frends Technology 2023

Antti Toivanen
Timo Vesa & Inga Miettinen

Table of Contents

Copyright Frends Technology Oy – all rights reserved.

Copyright Frends Technology Oy – all rights reserved.

1 Executive Summary

This white paper presents a comprehensive guide to effective integration delivery, rooted in the extensive 35-year expertise of Frends integration specialists. Central to the discussion is the "Integration Factory" framework, a versatile model designed to accommodate various delivery methodologies, from the classic waterfall to the agile Sprint and the specialized Artifact Driven Model. The framework's design ensures scalability, suitable for both compact teams and expansive multi-team structures.

Key pillars of the integration factory model, such as Integration backlog governance, architectural best practices, and role delineation, are explored in depth. The document also introduces the Agile Integration Model (AIM) and the Artifacts Driven Model (ADM), each tailored to address the distinct challenges inherent to integration projects. Foundational concepts, including the "Definition of Ready" (DoR) and "Definition of Done" (DoD), are spotlighted for their pivotal role in ensuring clarity, efficiency, and consistent quality in integration endeavors.

Concluding with an insight into Frends Technology's integration service offerings, this paper serves as a roadmap for partners and stakeholders, guiding them towards efficient, adaptable, and high-quality integration outcomes.

Copyright Frends Technology Oy – all rights reserved.

2 Integration Factory

Let's first have a look at the delivery framework we call "Integration Factory". The framework still allows multiple delivery models like old waterfall, agile Sprint model and integration maturity specific Artifact Driven Model.

Keep in mind that this framework can be built with only a few people or can include tens of people and even multiple teams.

Key concepts in integration factory model.

1. The Integration backlog is governed by the customer and followed according to the service governance model. More specifically, integration backlog can be divided into business backlog governed by customer and development backlog governed by delivery organization.

2. Integration design and management to ensure following the best practices set in the architecture guidelines and the integration handbook.

3. Integration development by a customer-specific / project-specific team or on flexible resourcing. Dedicated Project Manager named for larger projects. It is optionally a DevOps team doing maintenance work for existing integrations as production support takes care only of critical production incidents.

4. Production Operations team managing the production environment, incident & problem management, and SLA with proven processes for support and maintenance from over 100 customers.

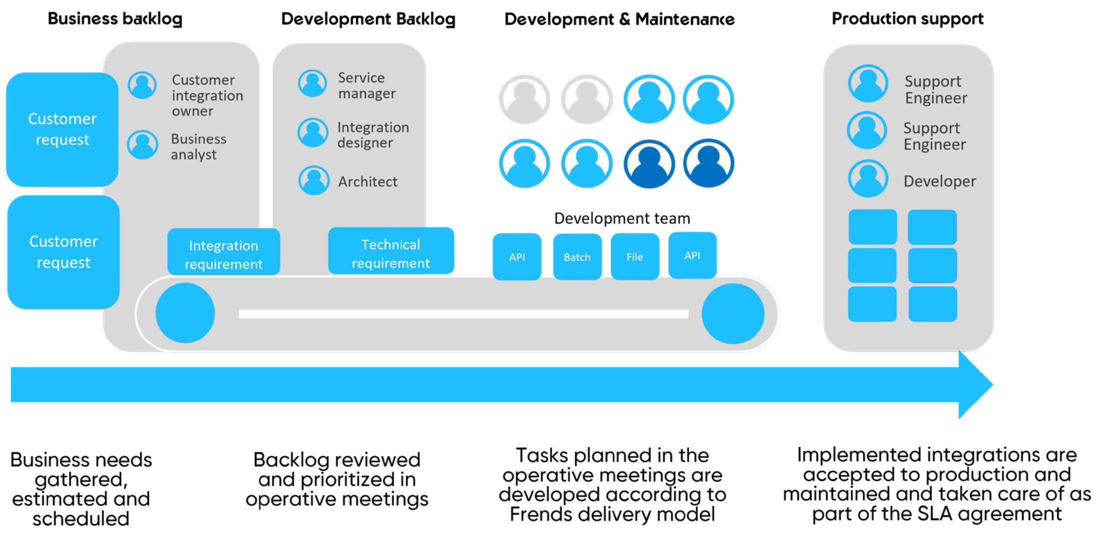

Business backlog
Development Backlog
Development & Maintenance
Production support
Customer request
Customer request
Customer integration owner
Business analyst
Service manager
Integration designer
Architect
Integration requirement
Technical requirement
Development team
API
Batch
File
API
Support Engineer
Support Engineer
Developer
Business needs gathered, estimated and scheduled
Backlog reviewed and prioritized in operative meetings
Tasks planned in the operative meetings are developed according to Frends delivery model
Implemented integrations are accepted to production and maintained and taken care of as part of the SLA agreement

Figure 1 Integration Factory Framework

In the factory model, Customer Integration owner jointly with Business analyst build and upkeep Business backlog containing business-oriented user stories. These stories are purged into Development Backlog as Integration Backlog Items by Business analyst (aka integration designer). This Development is managed by a Service Manager (Artifact Driven Model, Continuous development) or by a Project Manager depending on is the context project or continuous development. The same framework can serve both continuous delivery and projects at the same time – only staffing changes as larger projects need their separate project managers.

Copyright Frends Technology Oy – all rights reserved.

3 Integration Delivery Models

Due to the nature of Integration projects i.e., in contrast with software development project, we have identified two possible agile delivery models. Those are Artifact Driven Model and Sprint Model.

The Artefacts Driven Model is meant for smaller projects, usually lasting 60 days maximum, whereas the Sprint Model is for more than 120 days.

We found that, in contrast to software development projects, integration projects require a different approach in their *development* as well as *deployment*.

Before diving into delivery models, let's clarify a few essential concepts required for success and these delivery models: Artifacts, Definition of Ready, and Definition of Done.

3.1 Artifacts

Artifact, in this context, is the common name for essential requirements and documents needed for integration development.

Common integration artifacts are:

1. Integration Story – what does this specific integration do business-wise?
2. Functional Requirements – what functionalities are required to implement the user story?
3. Non-functional Requirements – what technical needs there are, for example, how many calls per second will approximately be for the API you are developing or how many executions there will be for process automation.
    a. Performance requirements
    b. Regulative requirements
    c. Integration response time maximum
    d. GDPR requirements
    e. Security approvals needed.
    f. Data protection approvals?
4. Description of Source and destination interfaces
    a. Include file paths, credentials, authentication e.g., for APIs, note that the credentials should not be visible on the backlog item.
5. Data-mapping charts or excels.
6. Common requirements
    a. Is archiving required?
    b. Is extended monitoring required?
    c. Error management, where should alerts go?
    d. Other matters to error management, e.g., retries or special exception handling for commonly known errors.
    e. How is integration triggered? File came, API call came, scheduling, change in e.g., SAP.
7. Sample messages and / or files or happy cases.
    a. Avoid "hand-made" samples, trust to real ones from real system interface.
    b. Never use production data if it may contain Person Identification Information (PII) or sensitive data.
8. Identified exception cases; it is not uncommon to exception handling take a bigger amount of development time than happy cases. This time is saved in production.
9. Testing requirements
    a. Source and Target system test environments

Copyright Frends Technology Oy – all rights reserved.

b. Developer testing responsibilities, e.g., automated unit tests
c. Business testing responsibilities

3.2 Definition of Ready (DoR)

The "Definition of Ready" (DoR) is commonly used in Agile software development, particularly Scrum methodologies. It refers to a set of criteria that a user story or product backlog item **must meet before it can be considered "ready" to be taken into a sprint for development**. The DoR ensures the team has a clear and shared understanding of what is expected before work begins, reducing ambiguities and potential rework.

The "Definition of Ready" is a powerful tool in the Integration delivery toolkit. It acts as a gatekeeper, ensuring that only well-defined and understood work enters the sprint, leading to more predictable outcomes and higher team efficiency. It's essential to note that the DoR should be flexible and can be adapted based on the team's experiences and needs.

3.2.1 Purpose and Benefits of the Definition of Ready

The goals of agreeing on DoR are the following:

- Clarity: Ensures that the team has a clear understanding of the work to be done.
- Efficiency: Reduces the time spent on rework or clarifications during the sprint.
- Predictability: Helps in estimating the work more accurately.
- Collaboration: Ensures that all stakeholders (e.g., product owner, developers, testers) have had input and agree on the criteria.

The benefits of DoR are:

- Reduced Waste: By ensuring stories are ready, teams can reduce the time spent on unnecessary tasks or rework. The most common symptom of missing DoR are iterations that create unplanned costs and ruin the schedule.
- Improved Communication fosters better communication between the product owner and the development team.
- Better Sprint Planning: Sprint planning meetings become more efficient and productive with ready stories.
- Higher Quality: Clear criteria lead to better end products as the team knows exactly what's expected.

3.2.2 Common Criteria Set in Integration Development

- User Story is Clearly Defined: The story has a clear description, and its purpose is understood.
- Acceptance Criteria: The expected outcomes and conditions for the story's completion are listed. For example, how is the integration case tested? Is there a

need for automated unit tests? This links to the Definition of Done – an integration is not "Done" until these acceptance criteria are met.

- **Dependencies Identified**: Any internal or external dependencies are known and can be managed.
- **Size**: The story is small enough to be completed within one sprint.
- **Feasibility**: The team believes the story can be completed with current knowledge and resources.
- **No Ambiguities**: Any doubts or questions about the story have been addressed.
- **Artifacts are there**: The work should not start before the artifacts are there. If it must start without artifacts, both parties should prepare a budget and schedule for one-to-many iterations depending on what artifacts are missing initially.

3.3 Definition of Done (DoD)

The "Definition of Done" (DoD) is another fundamental concept borrowed from Agile software development, especially within Scrum methodologies. It refers to a set of criteria a user story, task, or integration backlog item must meet to be considered "done" or "completed." The DoD ensures a shared understanding across the team about what it means for work to be finished, promoting consistency in quality, and ensuring that all necessary steps have been taken. More importantly, it ensures that the integration delivery team and the customer receiving integrations have a shared understanding when the integration case is ready.

3.3.1 Purpose of Definition of Done

The goals of agreeing on DoD are the following:

- **Consistency**: Ensures that all work meets a consistent standard of quality.
- **Transparency**: Clarifies stakeholders about what to expect from a completed item.
- **Quality Assurance**: Acts as a checklist to ensure all necessary quality steps are taken.
- **Shared Understanding**: Ensures that the entire team and customer have a mutual agreement on when a task is truly completed.

The benefits of having a clear DoD agreed upon between the delivery team and the customer are the following:

- **Quality Control**: Ensures that all completed integrations meet the team's quality standards.
- **Clear Expectations**: Provides a clear understanding for the team about what is expected for work to be considered complete.
- **Improved Deliverables**: Ensures that what is delivered to the customer or end-user is high quality and meets their expectations.

Copyright Frends Technology Oy – all rights reserved.

- Reduced Rework: By adhering to the DoD, teams can reduce the chances of having to revisit completed work in the future, thus reducing the amount of budget and schedule challenges.

3.3.2 Common Criteria Set in Integration Development

While the specific criteria can differ from one team to another, common elements in a DoD might include:

- Completion: The integration for the task or story is fully developed and checked into the version control system. Acceptance criteria set in Dor is met.
- Unit Review: The integration has been reviewed by peers for quality and maintainability.
- Testing: All necessary tests (unit, integration, system) have been written and passed successfully.
- Documentation: Any required documentation, whether code comments or external, has been completed.
- No Known Defects: The item has been tested, and no known defects remain.
- Stakeholder Approval: The product owner or relevant stakeholder has reviewed and accepted the work.

3.4 Agile Integration Model (AIM)

In AIM, the integration factory is in place, and the customer also has an integration architecture set in place. Integration architecture gives rules, guidelines, and restrictions on implementing technical requirements. Before AIM can start, DoR and DoD have also agreed between the customer and delivery party.

Depending on the number of integration backlog items, the supplier can allocate an appropriate number of developers and integration designers to handle the "integration lane." Note that supplier may establish Integration factory that serves multiple customers in separate Business Backlogs.

Copyright Frends Technology Oy – all rights reserved.

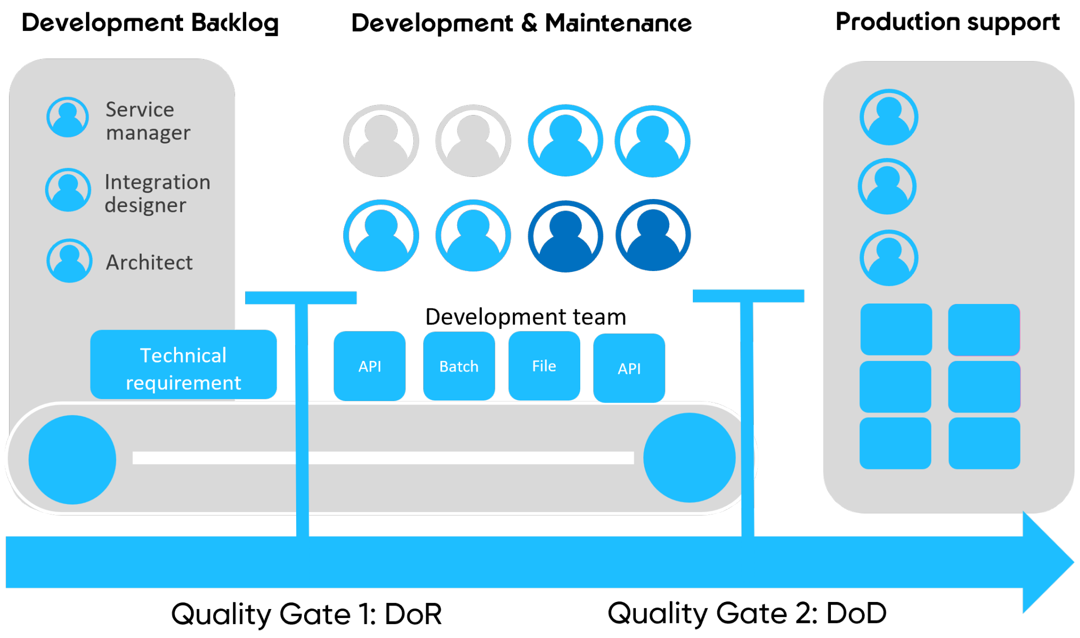
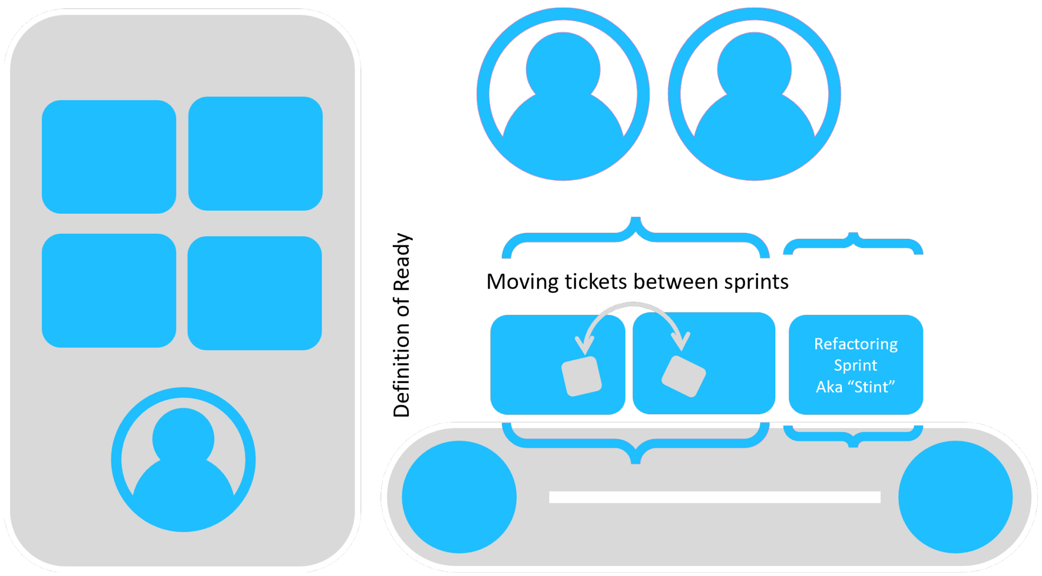


Figure 2 Integration Factory with DoR and DoD Quality gates present

3.5 Integration Projects

The AIM model is suitable for both continuous delivery and project delivery at the same time. The allocation of the Development team should always mirror the amount of backlog items there is – whether from projects or continuous development. We strongly recommend that quality gates DoR and DoD are present in all integration development.

Integration projects typically are the part that glues other systems together. It might be the ERP vendor that provides an interface that is not working as expected or the API of a SaaS that cannot handle the number of concurrent calls from the integration platform (a non-functional requirement). Additionally, artifact quality is often not good. All these risks can be taken into account with Stints. Stint is a name for an additional sprint reserved for surprises and situations that could not be seen beforehand. Stint might be a shorter version of a sprint. The amount and schedule of Stints in the overall project schedule should be based on the quality of artifacts and history with the suppliers.


Definition of Ready
Moving tickets between sprints
Refactoring Sprint Aka "Stint"

When having a project with lots of Integration Backlog items, usually the design work needs to start before the development sprint. In other words, design (artefact gathering) is on-going work for every sprint for forthcoming sprints. This also means that the first sprint may be a design sprint to get a set of DoR approved integration backlog items for the first sprint.

Overall view of a sample integration project may look like the following illustration. Remember that the top-level integration architecture might be done separately and sold as a separate service.

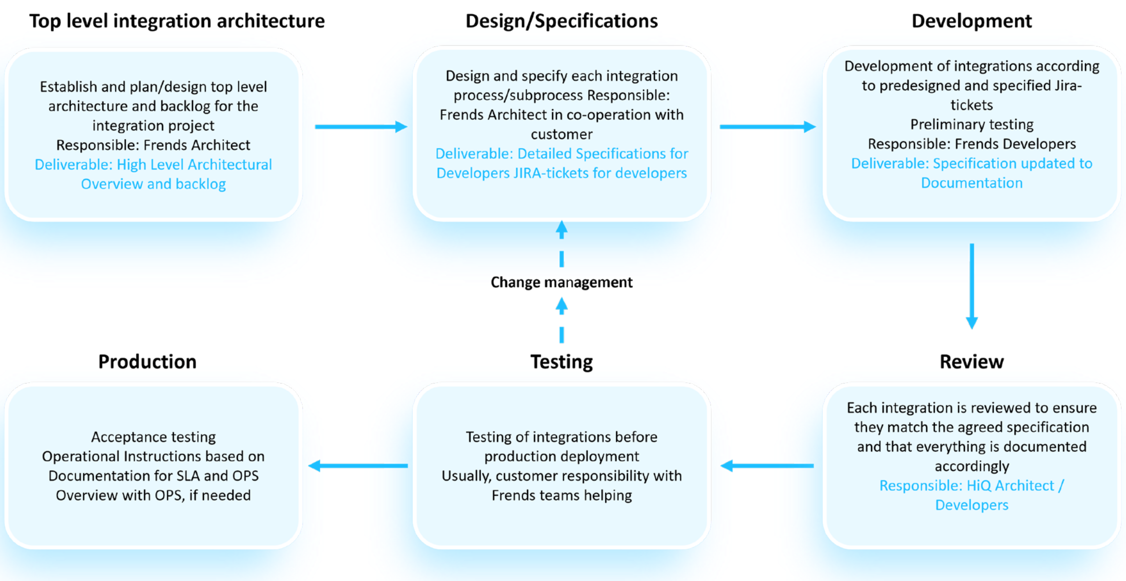

Top level integration architecture
Establish and plan/design top level architecture and backlog for the integration project
Responsible: Frends Architect
Deliverable: High Level Architectural Overview and backlog
Design/Specifications
Design and specify each integration process/subprocess Responsible: Frends Architect in co-operation with customer
Deliverable: Detailed Specifications for Developers JIRA-tickets for developers
Development
Development of integrations according to predesigned and specified Jira-tickets
Preliminary testing
Responsible: Frends Developers
Deliverable: Specification updated to Documentation
Change management
Production
Acceptance testing
Operational Instructions based on Documentation for SLA and OPS
Overview with OPS, if needed
Testing
Testing of integrations before production deployment
Usually, customer responsibility with Frends teams helping
Review
Each integration is reviewed to ensure they match the agreed specification and that everything is documented accordingly
Responsible: HiQ Architect / Developers

Figure 3 Sample Project Delivery Functions

3.6 Artifacts Driven Model (ADM)

This approach is meant for smaller projects, usually 60 days max. It is also feasible for continuous work where the need for integration drops in now and then. The artifact-driven model is applicable for continuous development when the customer maturity level does not allow all artifacts to be present when work starts, yet the work needs to start. ADM is not the number one choice for delivery and if you are forced to go there, always try to establish more mature design process where the missing artifacts are identified, and iterations planned schedule and cost-wise.

The model reflects that artifacts are usually collected in an ad-hoc style and might not be ready at the beginning of the implementation work. Establishing DoR and DoD is still crucial as they make customers understand when initiating work that might need iterations to be completed. This way, the delivery party and customer can plan time and budget for iterations together beforehand, not by arguing afterward.

Commonly, organizations do not understand what artifacts are needed for integration. This leads to the organization asking for artifacts from third parties, for example, SAP suppliers, one artifact at a time. In addition, it's quite common that an organization cannot evaluate the quality of artifacts.

Copyright Frends Technology Oy – all rights reserved.

The approach takes into account the fact that, in integration projects, very often, artifacts are neither delivered in full nor correctly. This reflects the complexity inherent in most integration projects.

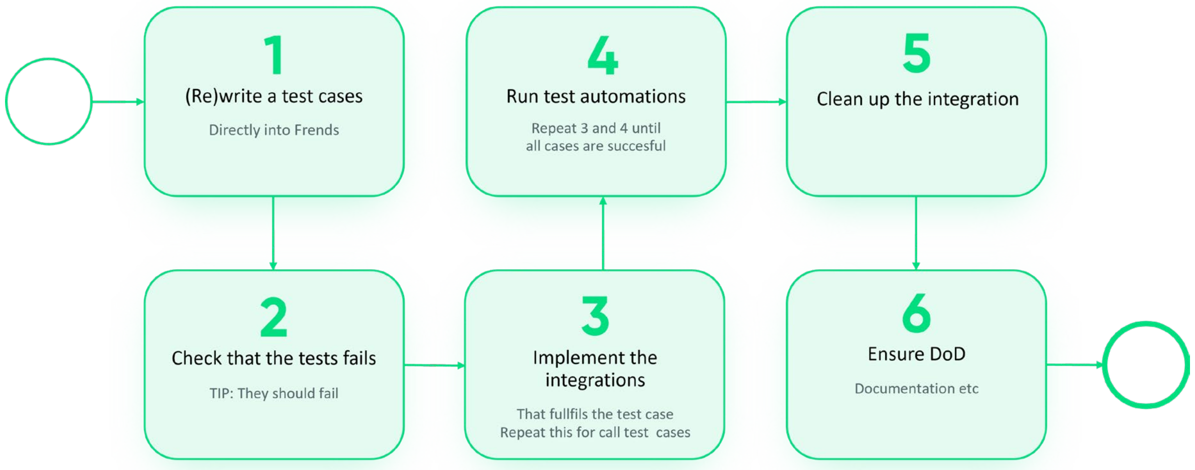
Planning a Stint for every integration backlog item in ADM is a good idea. Remember to include the added work amount and impact on the overall schedule in that plan.

3.7 Test Driven Development

Test-driven Development (TDD) is one of the best ways to ensure quality of unit testing in integration development. The idea is to first create automated test cases. The first test round should fail. This failure occurs because the developer has not yet implemented the integration solution that leads to success and ensures that test cases work correctly.

A developer can start his or her work after all paths of process execution have been covered with automated test cases. Each test case should turn green to illustrate that the integration solution under construction is coming closer to its ready status.

Acceptance Test Driven Development is an advanced version of TDD, where Test case is the acceptance test case.


1
(Re)write a test cases
Directly into Frends
2
Check that the tests fails
TIP: They should fail
3
Implement the integrations
That fullfils the test case
Repeat this for call test cases
4
Run test automations
Repeat 3 and 4 until all cases are succesful
5
Clean up the integration
6
Ensure DoD
Documentation etc

4 Roles in Integration Delivery

This chapter describes the roles and responsibilities of Integration Factory style delivery. Note that these roles can be combined in a way that one person has two roles. Avoid the "one-man band" approach where one person is responsible for artifacts and development as it will lead to a challenge where on person is too much responsible of single integration case. This will eventually lead to a situation where maintenance is hard during vacations and sick leave.

4.1 Integration Architect

An Integration Architect is responsible for designing and implementing enterprise-level integrations between software applications and systems. They ensure that different software products, often from various vendors, work seamlessly together to achieve

Copyright Frends Technology Oy – all rights reserved.

business objectives.

4.1.1 Key Responsibilities

Design and Development: Create integration blueprints to ensure that different software systems communicate effectively. This includes defining the structure of data, mapping data between systems, and determining the best integration patterns to use. Integration architects also set the rules for integration development. These rules usually include integration naming conventions, version control, and generic base roles for Definition-of-Ready and Definition-of-Done.

Scalability and Redundancy: Design integration solutions that can handle increasing data volumes and ensure that there are backup and failover mechanisms in place.

Analysis: Work closely with business stakeholders and IT teams to understand the business requirements and translate them into technical specifications.

Integration Tools: Utilize integration platforms and tools such as middleware, ESB (Enterprise Service Bus), API gateways, and others to facilitate integration.

Collaboration: Work with software developers, system architects, and other IT professionals to ensure that the integration solutions are aligned with the overall IT strategy and architecture.

Performance Tuning: Monitor the performance of integration points and make necessary adjustments to ensure optimal performance and scalability.

Documentation: Create detailed documentation for integration designs, data flow diagrams, and other relevant technical details.

Troubleshooting: Address any integration issues that arise, working closely with the relevant teams to diagnose and resolve them.

Continuous Improvement: Stay updated with the latest integration technologies and best practices to ensure that the organization's integration architecture remains modern and efficient.

4.1.2 Skills and Qualifications

Technical Proficiency: Deep understanding of integration patterns, API design, and middleware technologies.

Analytical Skills: Ability to analyze complex technical challenges and come up with effective integration solutions.

Communication: Strong communication skills to convey technical concepts to non-technical stakeholders.

Complex Problem-Solving: Ability to troubleshoot and resolve integration challenges.

Knowledge: Familiarity with various integration tools and platforms, as well as industry standards like REST, SOAP, and others.

AI-knowledge: For Hyperautomation cases the skills to use e.g. Azure Cognitive Services and Machine Learning services (or corresponding tools from other cloud platforms) is useful.

Copyright Frends Technology Oy – all rights reserved.

4.2 Service Manager / Integration Manager

Service Managers manage the service produced for the customer. Those services usually are continuous development (small new features and changes to existing ones) and support services like production incident management. Note that if there is a larger integration project, those have dedicated Integration Project Managers that manage the development backlog of the project.

4.2.1 Key Responsibilities

Manage the "conveyor belt" of integration backlog items: Service manager ensure that the backlog items fulfill their DoR, get implemented by developers and meet the DoD and end up into the production agent successfully.

Communications: Service Managers arrange the weekly and monthly meetings with the customer and the development team. Service Managers communicate with the customer business backlog owner.

Coordinate the design: Service Managers coordinate the work of Business analysts together with the customer.

Staffing: Service Managers ensure that every part of the integration, delivery and support process is staffed correctly. Note that in factory delivery model the amount of business analysts and integration developers must match the backlogs. For example, if development backlog is quite empty but business backlog is full of items, Service Managers must take staffing actions and initiate the artifact gathering early to ensure that the delivery process runs efficiently.

4.2.2 Skills and Qualifications

Communication: Strong communication skills to convey technical concepts to non-technical stakeholders.

Management skills: Excellent management skills and understanding of the tooling like JIRA or other work management tools.

Analytical Skills: Ability to analyze complex business and human challenges and come up with effective delivery.

Proactive mindset: Ability to tackle problems before they become a problem.

Technical Proficiency: Brief high-level understanding of integration patterns, API design, and middleware technologies.

4.3 Business Analyst / Integration Designer

Business Analyst, also known as Integration Designer, gathers the items to Business Backlog with the customer. She or he is responsible for adding the required artifacts to Business Backlog Items so that they fulfill their Definition-of-Ready.

Copyright Frends Technology Oy – all rights reserved.

4.3.1 Key Responsibilities

Backlog items: Business Analysts fill in the information to backlog items so that implementation can start. All the artifacts agreed in Definition-of-Ready are the responsibility of Analyst. If a separate business backlog is used, Analyst fills in the artifacts needed.

4.3.2 Skills and Qualifications

Business point of view: Business analysts must have a process-oriented mindset and ability to grasp the customer's processes – business and technical – rapidly.

Technical Proficiency: Understanding of integration patterns, API design, and middleware technologies.

AI-knowledge: For Hyperautomation cases the skills to use e.g. Azure Cognitive Services and Machine Learning services (or corresponding tools from other cloud platforms) is useful.

4.4 Integration Developer

Developers are the core part of the integration team implementing the backlog items.

4.4.1 Key Responsibilities

Development: Understand and implement backlog items and ensure they meet their Definition-of-Ready. In addition to DoR, implemented items must meet the general requirements described in an integration architecture plan or handbook.

Testing: Conduct unit, integration, and system testing to ensure that the integration solutions work as expected. Implement automation tests when needed.

Deployment: Deploy integration solutions to production and other environments, ensuring that they are scalable and performant.

Monitoring and Maintenance: Monitor integration solutions for any issues or failures. Address any bugs or issues that arise post-deployment.

Documentation: Create detailed documentation for the integration solutions, including design specifications, data mappings, and user guides.

Performance Tuning: Optimize integration solutions to ensure they meet performance benchmarks and reduce latency.

Security: Ensure that all integrations adhere to security best practices, including data encryption and secure data transfer.

Collaboration: Work closely with other IT teams, such as software developers, database administrators, and infrastructure teams, to ensure seamless integration.

Continuous Learning: Stay updated with the latest integration technologies, tools, and best practices. This includes understanding new integration patterns and methodologies.

Copyright Frends Technology Oy – all rights reserved.

4.4.2 Skills and Qualifications

Programming oriented: Whilst modern integration platforms like Frends iPaaS do not require deep programming skills, understanding of the basic concepts like loops and if-then-else branches is required.

Process automation understanding: Understanding the process level automation is essential. This requires a more general understanding of the overall automation at hand.

API Development: Knowledge of RESTful and SOAP web services.

Data Formats: Familiarity with XML, JSON, and other data interchange formats. Additionally, knowledge of EDI, UBL and specific standards are required.

Databases and query methods: Understanding of relational and NoSQL databases. Knowledge of SQL, gRPC and GraphQL are generally required.

Version Control: Proficiency with tools like Git or SVN.

Test-Automation: Frends includes option to create automated unit test. Skills to create proper test cases are essential.

AI-knowledge: For Hyperautomation cases the skills to use e.g., Azure Cognitive Services and Machine Learning services (or corresponding tools from other cloud platforms) is useful.

5 Ingredients of Successful Integration Delivery

This chapter contains checklists for integration delivery parties to ensure successful customer integration projects and continuous delivery.

5.1 General and Continuous delivery

1. **Avoid** starting development **without Artifacts** present.
2. Always agree on **DoR** and **DoD** beforehand
3. Be aware that work estimations of each individual backlog item **must always include management, integration design, integration development, testing / test automation and production deployment**.
4. Starting development without all artifacts leads to iterations, if you are **forced** to **start** for example because error and trial is the only way to find missing artifacts, **plan schedule** and **cost** estimations accordingly.
5. **Avoid "one-person band"** staffing where one person does e.g., the design and development.
6. If artifacts are not present and the iterations are the only way, plan costs and schedule accordingly.
7. **Use Test Driven Development or Acceptance Test Driven Development** when those test cases are available before implementation starts.

5.2 Integration Projects

Copyright Frends Technology Oy – all rights reserved.

1. All best practises from previous chapter of general and continuous delivery.
2. Ensure that the project has **proper roles present**.
3. **Ensure** that **all the stakeholders,** including customer business owner and other supplier of systems that will be integrated, are involved.
4. Even if the integration project is commonly a subproject of a larger entity, like ERP project, remember that **integration project management is a skill of its own.** Assign an experienced PM with integration understanding.
5. Estimate the **quality and maturity of Artifacts, plan sprints, stints, and costs accordingly** right from the start of the project.

Copyright Frends Technology Oy – all rights reserved.

6 Frends Service Mix

Delivery of integration solutions if not of course the whole story. Before implementing integration to be onboarded. After implementation of the first integration project, customers need maintenance and support. Following services play key role in overall integration service to end customer:

6.1 Onboarding

Onboarding is a crucial step. It can pave the path to high-value low TCO integration or vice versa. Ensure, that at onboarding phase:

- the business requirements and needs are considered.
- the naming conventions, documentation, and general rules (a base set for DoR and DoD) are established.
- the right people are trained (free online Frends courses for example) and certified.
- The best practices are taken into use by trained, motivated and skilled people.

Quite often the business requirements, deployment architecture of integration platform, basic processes and the best practices are set in integration architecture plan or it's subset Integration handbook.

6.2 Delivery

Integration Delivery can be done with any old or new IT delivery method. From delivery organization perspective those can be:

1. Consultant for hire, where supplier rents the needed roles for customers.
2. Waterfall (not described in this document, yet methods and roles from this whitepaper can be used.)
3. Agile Continuous Development and Project delivery – the Integration Factory and Agile Integration Model described in this whitepaper.

6.3 Support

Integrations play an essential role in the core business process automation and are commonly business critical. Almost all the integration customers require support and maintenance, quite often with service level agreement backed reaction and resolution times. Whilst a minor set of customers handle support themselves, most want the delivery party to support their own deliveries. This is a great opportunity for delivery organizations to broaden the offering from mere delivery. Quite often is an integral part of DevOps and DevSecOps styled Integration Factory delivery.

If your own delivery organization cannot offer support services, for example the 24 / 7 reaction service, you can always ask Frends Organization to establish joint offering.

Copyright Frends Technology Oy – all rights reserved.

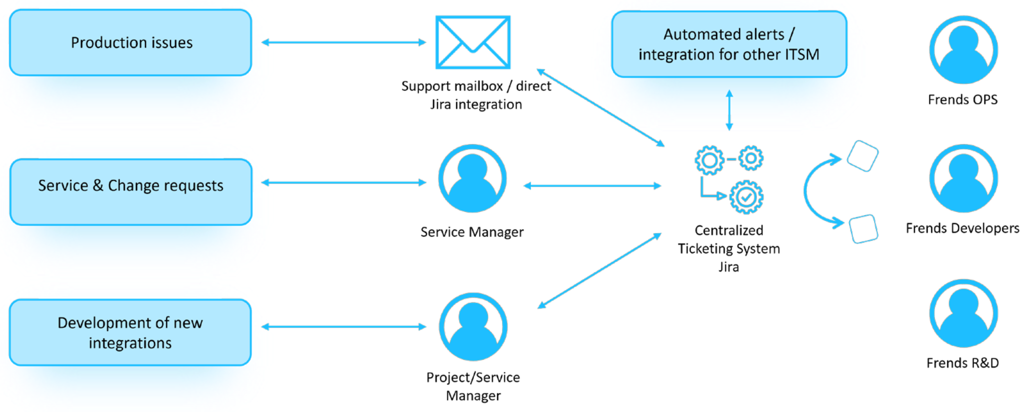

Production issues
Service & Change requests
Development of new integrations
Support mailbox / direct Jira integration
Service Manager
Project/Service Manager
Automated alerts / integration for other ITSM
Centralized Ticketing System Jira
Frends OPS
Frends Developers
Frends R&D

Figure 4 Sample Support Organization and Tooling from Frends

6.4 Frends Technology joint offering

The following illustration shows how we at Frends Technology offer integration services. As a partner you might want to consider a suitable service mix for you are offering. Remember, you can always create your offering with us – for example you can agree what we do the initial architecture for your customer or take over the support services if you are not able to implement these parts yourself.

Onboarding Services
Architecture guidelines – Fixed fee
Integration handbook to ensure development quality – Fixed fee
Frends Academy – Fixed one-time fees

Consultants for hire

Consultants and developers available for Customer projects on fixed allocation. Resource management by the Customer.

Frends continous Delivery

Flexible or fixed resource allocation based on Customer backlog. Resource management by Frends.

Project Deliveries

Project deliveries and resources for specific project scope. Resourcing agreed per project.

Support Service

Solution support according to agreed SLA.
Support work included in fixed support fee or T&M.

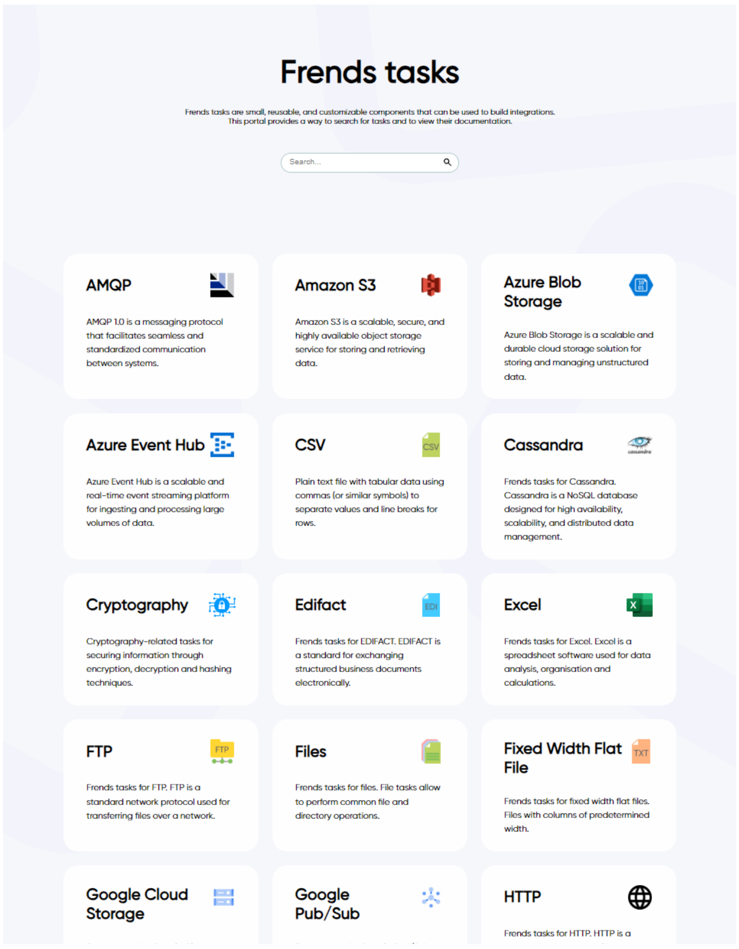
6.5 Marketplaces

Frends offers ready-made connectors and prepackaged integration processes (PiPs) at tasks.frends.com.

If you or your team has something to share, participate in the community. You can also sell and distribute your Frends PiPs and Connectors in this marketplace soon.

Frends is also available on third-party marketplaces like Azure Marketplace and Acronis Marketplace and many more. If you and your company has own marketplace, you can always add Frends there. This is

Copyright Frends Technology Oy – all rights reserved.



Frends tasks

Frends tasks are small, reusable, and customizable components that can be used to build integrations.
This portal provides a way to search for tasks and to view their documentation.

Search...

AMQP
AMQP 1.0 is a messaging protocol that facilitates seamless and standardized communication between systems.

Amazon S3
Amazon S3 is a scalable, secure, and highly available object storage service for storing and retrieving data.

Azure Blob Storage
Azure Blob Storage is a scalable and durable cloud storage solution for storing and managing unstructured data.

Azure Event Hub
Azure Event Hub is a scalable and real-time event streaming platform for ingesting and processing large volumes of data.

CSV
Plain text file with tabular data using commas (or similar symbols) to separate values and line breaks for rows.

Cassandra
Frends tasks for Cassandra. Cassandra is a NoSQL database designed for high availability, scalability, and distributed data management.

Cryptography
Cryptography-related tasks for securing information through encryption, decryption and hashing techniques.

Edifact
Frends tasks for EDIFACT. EDIFACT is a standard for exchanging structured business documents electronically.

Excel
Frends tasks for Excel. Excel is a spreadsheet software used for data analysis, organisation and calculations.

FTP
Frends tasks for FTP. FTP is a standard network protocol used for transferring files over a network.

Files
Frends tasks for files. File tasks allow to perform common file and directory operations.

Fixed Width Flat File
Frends tasks for fixed width flat files. Files with columns of predetermined width.

Google Cloud Storage

Google Pub/Sub

HTTP
Frends tasks for HTTP. HTTP is a

Figure 5 tasks.frends.com

To have a contributor access, contact your partner manager or sales@frends.com.

7 ISV Partners

ISV Partners is a Frends Partner that have their own product, for example a SaaS, that they offer to their customers.

Integrations are the biggest headache for any SaaS vendor. Frends iPaaS can be as OEM or white-labelled as a part of your offering.

With Frends, SaaS Vendors can focus on what matters most. Developing their core business, being innovative and creative and driving customer value. Focus on what matters most.

1. Scale up your SaaS core business.
2. Focus on Developing your product, not the connectivity.
3. Ensure customer satisfaction.
4. Adapt to customer needs.
5. Integrations made fast = Reduce time from closed deal to invoicing

Copyright Frends Technology Oy – all rights reserved.

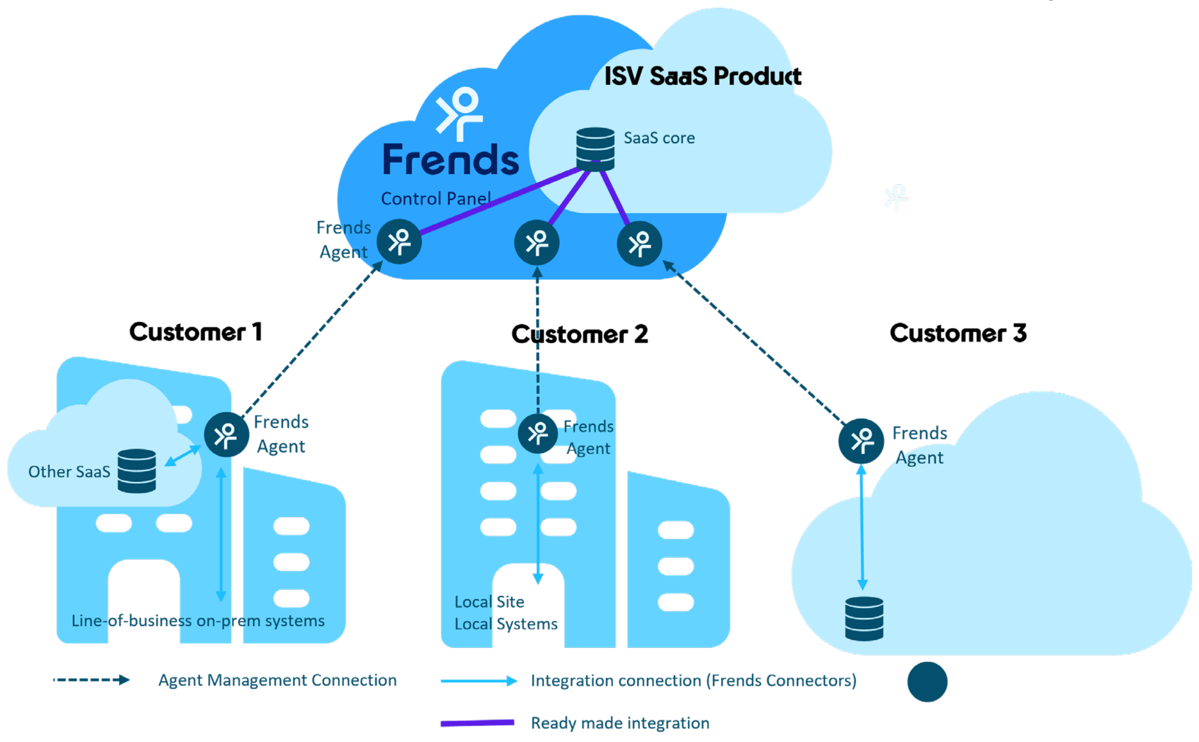


Figure 6 ISV framework, connections to ISV's SaaS are ready made, just connect the customer end.

In ISV framework, the integrations from Frends to ISV's SaaS are made. It could be considered as a set of cables (integrations) where the other end is connected, and the other end is connected to each new customer as they arrive.

To ISV to be able to connect one Frends Installation to many customers, a specific ISV contract with Frends Technology Oy is required.

ISV Partner may establish an Integration Factory and use the same delivery methods and best practices as System Integrators.

Copyright Frends Technology Oy – all rights reserved.

8 Summary

This white paper delves into the intricacies of integration delivery, drawing from the vast 35-year experience of Frends integration specialists. At its core, the document introduces the "Integration Factory" framework, adaptable to various delivery models like the traditional waterfall, agile Sprint, and the integration maturity-specific Artifact Driven Model. The framework's flexibility allows for scalability, accommodating both small teams and larger multi-team setups.

Key components of the integration factory model include the governance of the Integration backlog, adherence to architectural guidelines, and the division of roles among development, management, and production operations teams. The document further elaborates on the Agile Integration Model (AIM) and the Artifacts Driven Model (ADM), both tailored to address the unique challenges of integration projects. Essential concepts like "Definition of Ready" (DoR) and "Definition of Done" (DoD) are highlighted, emphasizing their role in ensuring clarity, efficiency, and quality in integration delivery.

The paper concludes with an overview of Frends Technology's service offerings in the integration domain, suggesting potential collaboration avenues for partners. As integration projects often present unforeseen challenges, the proposed models and guidelines aim to streamline the process, ensuring efficiency, adaptability, and high-quality outcomes.

If you require help, please contact your service manager or partner manager or sales@frends.com

About Frends

Integrations made easy.

Frends is an Integration Platform as a Service (iPaaS) designed to simplify integrations, reduce costs, and enhance efficiency, allowing businesses to focus on growth and customer satisfaction. The platform offers features such as visual design of integrations using the globally recognized BPMN 2.0 standard, low-code configuration, multicloud and hybrid capabilities, and compliance with cloud integration standards. Frends has received accolades and certifications from various organizations:

G2: Recognized as one of the best integration platforms, Frends has earned leader badges for "Easiest Setup", "Fastest Implementation", and "Easiest Admin".

Gartner Peer Insights: Users have rated Frends iPaaS with an impressive 4.7 out of 5.

ISO Certifications: The International Organization for Standardization (ISO) has qualified Frends for ISO 9001, ISO 14001, and ISO 27001, indicating its commitment to quality, environmental management, and information security.

Nixu Certification: A cybersecurity company, Nixu has certified Frends, ensuring that its development procedures prioritize cybersecurity and adhere to best practices.

The platform has been adopted by various sectors, including transportation, healthcare, and media & entertainment, showcasing its versatility and effectiveness in different industries.

Read more at frends.com