

## Integrating Mavenlink and SAP S/4 HANA – Business and Technical Consideration



# CONTEXT

Mavenlink and SAP S/4 HANA are both commercial systems that automate business processes. While Mavenlink is a project management solution, SAP S/4 HANA rolls out entire ERP solutions. This paper will outline the considerations for integrating a Mavenlink system with SAP S/4 HANA. Specific niche cases will not be discussed as these are contingent on a wide range of potential business requirements, but the general business case is to capture data for financial processes and reporting from Mavenlink into S/4.

Mavenlink offers a variety of functionalities for project management, workflows, task tracking, resource management, project reporting and project accounting. When S/4 is the business's ERP system of choice, SAP Project Systems (PS) is used to seamlessly exchange data between Mavenlink and S/4. PS integrates directly into SAP financial and reporting modules, allowing users to view and manage project financials. S/4 consumes this data through Mavenlink's public API, authorized by OAuth 2.0 security protocol.

The overall limitation of both systems becomes the primary challenge for the team working on this integration. The team should address the full set of requirements around all edge cases upfront to avoid any rework. These cases should be documented and incorporated into the functional/technical specifications as early on as possible.

# CONSIDERATIONS

## For the Business Team

The team can use Project Systems (PS) to consolidate project and Work Breakdown Structure (WBS) related data from Mavenlink. A Work Breakdown Structure is simply the hierarchy of tasks, milestones and deliverables that make up a project. The elements of the WBS hierarchy in S/4 capture transactional information from Mavenlink. This information can flow into other modules such as Sales & Distribution (SD) and SAP Analytics Cloud (SAC). SD maintains the Sales Orders associated with the projects and receives project billing and revenue recognition information from PS. The business can leverage SAC to report on a project's profitability, resource utilization, and other attributes pulled from Mavenlink.

The integration team should understand how the business will consume the financial data of the projects coming into SAP. For example, some of the transactional information may need to behave differently based on project type – a milestone project recognizes revenue and invoices the customer differently than a prepaid project does, Free of Charge work should not result in billing, internal projects may only capture costs, etc. These cases should be discussed and documented.

Resource management is another consideration for this integration. Mavenlink offers resource management capabilities including resource assignment, resource utilization, time recording and timesheet approval workflows. Rate cards can be used to set default rates for roles across projects, both billing and cost rates. The integration team should consider how the business will use these capabilities in Mavenlink, as well as how the business manages internal resources as opposed to external resources.

If the business has existing data migrated into Mavenlink there's a chance they will need to pull that into S/4 as well. PS can accept historical data via file upload. If doing the migration via file upload, the team will need to clean the data prior so that S/4 accepts the format. The team can separate the data to upload in batches. This

isolates any issues that occur during the upload.

PS consultants, business analysts, ABAPers, SD consultants, and business SMEs will need to work closely to successfully implement this integration. The following paragraphs will dive deeper into some of the technical considerations required for the Mavenlink-Project Systems integration.

## For the Technical Team

The integration requires ABAP development to pull data from Mavenlink's public API and store in S/4. The API utilizes OAuth 2.0 security - a Mavenlink administrator will need to register the integration account in Mavenlink to receive the security token. SAP can consume Mavenlink's data model via GET, POST, PUT, and DELETE endpoints. The API response is always JSON. ABAP will need to deserialize and parse the response to process the data needed for the integration. Documentation for the API can be found here:

<http://developer.mavenlink.com/beta/>

Developers will need to pull Mavenlink projects (workspaces) and tasks (stories) into Project Systems to create and update the project definitions and corresponding Work Breakdown Structures. The business may find other areas of the data model relevant such as timesheets, time entries, rate cards, custom fields, and custom field values. Transactional objects such as timesheets can be pulled and processed in a separate executable. Developer(s) should separate the processing logic of creating the objects in S/4 for the first time from updating these objects. The order that these objects are pulled will make a difference. If a task is pulled before the corresponding project is created, for example, then the system will throw an exception.

User events will mark objects in the Mavenlink data model for creation/updates when the API is pulled. The integration team may find it useful to map user events in Mavenlink to API endpoints. As an example, adding a new task on a project would update Mavenlink's data model. The API would reflect the new state of the data model on a request from SAP to the corresponding workspace/story objects. The team can perform this exercise to delineate use cases and relevant user interface triggers for integration.

Developers can minimize the number of requests sent to the server by taking advantage of request parameters. The "include" parameter is offered by many endpoints to pull associated objects in a request. For example, the program can issue a request for story objects (tasks) and include all corresponding workspaces (projects). SAP will likely not require every single instance of an entity on every request to an endpoint. To filter the number of objects returned, developers can pass a datetime stamp to "created\_after" or "updated\_after." These parameters will pull only the objects that had been created/updated since the datetime stamp.

It's worth noting that Mavenlink paginates its responses. If the number of objects returned exceeds 20, then the "per\_page" parameter should be set to its non-default value. This parameter maxes out at 200. If the response exceeds the value of this parameter, then the code will need to loop through the pages to process all the objects.

The ABAP source code will need to parse the multi-level JSON response from Mavenlink. The developer(s) may choose to use field symbols to do this. Some of the BAPIs relevant to this integration may include the following:

- BAPI\_BUS2001\_SET\_STATUS to change the project status
- BAPI\_BUS2054\_SET\_STATUS to change the WBS status
- BAPI\_BUS2001\_CREATE to create a project

- BAPI\_BUS2001\_CHANGE to update a project
- BAPI\_BUS2054\_CREATE\_MULTI to create a WBS hierarchy
- BAPI\_BUS2054\_DELETE\_MULTI to delete WBS elements
- BAPI\_PROJECT\_MAINTAIN to create or delete a milestone
- BAPI\_PROJECT\_MAINTAIN to update a milestone
- BAPI\_ACC\_ACTIVITY\_ALLOC\_POST to post transactions
- BAPI\_GOODSMVT\_CREATE to post goods receipt for a purchase order

## CONCLUSION

Integrating SAP with a third-party system is not trivial and usually requires involvement from multiple areas of the business. Involving the relevant resources from each area early in the project will be vital for the team to capture the full range of requirements. Defining these requirements sets the cadence for a successful implementation from conception to the start of support.

Maintaining good business documentation around the Project Management practice will assist the team in identifying the edge cases that the integration needs to accommodate. A seasoned PS consultant is likely required to configure the PS module from its out-the-box state into a more versatile system that can communicate with Mavenlink and other S/4 modules as needed. This professional will need to work closely with an ABAP developer as well as analysts and SMEs.

### About Aptimized

Aptimized are a thought leadership company within many sectors. Its model ensures that seasoned experts with real life experience in the field manage and advise any industry sector.

The SAP practice boosts experts familiar with configuration, deployment, integrations, and maintenance of SAP S/4 HANA. As a certified SAP partner with combined experience of over 100 years in leadership roles, the knowledge of the technology industry means objective guidance and advice to enable efficiency and modernization.