

Master 2/Engineer Internship

Event-driven API Orchestration Architecture

How can we make event composition in an Event-Driven Architecture as easy as message composition in a Message Oriented Middleware?

1 Internship context

Numerous public and private Application Programming Interfaces (APIs) are available today: Web APIs turn the Internet into an application development platform, community APIs empower software ecosystems, and company-internal APIs modularize packaged software products or cloud services. Modern application frontends consume such APIs from different providers to compose a rich end user experience; Service-Oriented Computing and Event-Driven Architectures are two modern paradigms that enable asynchronous, API-driven Enterprise Application Integration (EAI) in software ecosystems, application platforms, and software-as-a-service offerings in the cloud.

Web API design is well understood and mature tools for API Design and Management exist on the market. However, the composition of API calls into end-to-end applications serving human user has not been investigated sufficiently yet; a number of patterns, methods, and tools from different communities exist. Business Process Management, for instance, can contribute workflow languages and engines, and the Domain-Driven Design community promotes Event storming, an agile workshop technique to rapidly craft service compositions and workflows. Powerful open-source frameworks for EAI such as Apache Camel allow system integrators to define long running, event-driven process flows without deep programming skills. Despite this rich set of candidate technologies and products supporting them, the orchestration (i.e., sequencing and/or parallel execution) of API calls in end user applications and intermediaries still requires a lot of human activity and error-prone work.

This deficit in the state of the art and the practice is particularly problematic for software vendors whose product portfolios comprise of highly distributed, configurable end user applications that works with heterogeneous data sources and external third-party systems. Such vendors often wish to provide their software services and product capabilities in a flexible, event-driven fashion. In such setting, it is not always feasible to find the deep expertise required to implement, test, and maintain the API call orchestrations in general-purpose languages such as Java or C#.

The study will be based on a given industrial process where events from two different applications must be joined in an easily customizable way in order to produce higher-value events and software operations afterwards. This would need to take into account ease of orchestration rules and event processing customization (through existing GUI or platforms, for example), but also to reflect on the associated challenges (for example, how long should we retain an event waiting for a complementary one to fire a new composite one?)

2 Company

This internship is proposed by SALVIA Développement, a French software editor based in Aubervilliers and composed with 100 persons (160 persons in Groupe SALVIA, adding the two subsidiaries ALTEVA and SRCI). SALVIA provides software for real-estate management (both private and public) during the whole lifecycle of the operations (finance, building follow-up, commercialization, maintenance).

The intern will be under direct supervision of Jean-Philippe Gouigoux, CTO of Groupe SALVIA, but will work in very close collaboration with other supervisors, as the subject is one of research on which SALVIA and the other participants have already exchanged on. Thus, the contacts of the intern will be multiple and in several locations. Work will be 100% remote, except for possible travel to Nantes or Paris, depending on the location of potential physical meetings.

3 Supervisors

1. Olaf Zimmermann, The Open Group Distinguished (Chief/Lead) IT Architect. Co-Autor (and master thesis advisor) of [Context Mapper](#); author of [MDSL](#) language and tools; master thesis advisor for [AsyncMDSL](#). See <https://ozimmer.ch/about/> for more information.
2. Jean-Philippe Gouigoux, CTO Groupe SALVIA, France, jp.gouigoux@salviadeveloppement.com
3. Dalila Tamzalit, Associate-Professor, LS2N, Université of Nantes, NaoMod team, expertise in software architecture evolution, Dalila.Tamzalit@univ-nantes.fr
4. Rafael Capilla, Associate Professor at Reu Juan Carlos University (URJC) of Madrid (Spain), Contact him at: rafael.capilla@urjc.es

4 Duration

6 months, starting end of January 2022 to end of July.

5 Objectives of the internship

The goal of this internship is to develop concepts, and a prototypical realization of them, for *event-driven API orchestration* including *pattern-based business event aggregation*. Relevant methods, tools and assets should be researched and compared w.r.t. requirements and criteria catalog derived from the above internship context. The exact scope of this evaluation activity will be defined jointly (students, advisors) at the start of the internship; candidate assets include those from above, i.e., event storming/BPM, Domain-Driven Design, Enterprise Integration Patterns [HohpeWolf 2003] and their Apache Camel implementation, as well as domain-specific languages such as CML and MDSL. The exact scope of this evaluation activity will be defined jointly (students, advisors) at the start of the internship; candidate assets include those from above, i.e., event storming/BPM, Domain-Driven Design, Enterprise Integration Patterns [HohpeWolf 2003] and their Apache Camel implementation, as well as domain-specific languages such as CML and MDSL.

The following questions should be answered:

- How can software product managers, application software specialists, and integration developers be supported with integration Domain-Specific Languages (DSLs) and tools so that they can call APIs and coordinate multiple API calls with ease when realizing distributed business capabilities? Can existing languages be extended for the given context, or are new ones required (and why)?
- How can logical branching (OR, XOR), parallel execution (AND), and join-merge aggregations in control flows be derived from business models (such as event storming outputs and BPM models) and expressed in a lightweight integration DSL? Which type of notation can serve the target audience better, text-based or graphical ones? How would a tool look like that exposes such an API orchestration notation to users?
- How do the developed concepts for event-driven API orchestration and their prototypical implementation compare to general-purpose programming of Web and messaging clients in Java and C# (in terms of learning effort, productivity, and cost of maintenance and evolution)?
- How much automation can be achieved with the proposed solution estimating effort saving and increase productivity?"

Critical Success Factors are:

- Developer Experience (DX): ease of configuration, tool usability
- Platform-independence and transferability of concepts
- Software engineering quality and adequate application of agile practices

6 Key milestones

1. Milestone #1: Compared the state of the art and technologies w.r.t. requirements and criteria in the application scenario (1 month).
2. Milestone #2: Proposed at least two alternative solutions, compare them, and recommend one for further elaboration (1 month)
3. Milestone #3: Designed and implemented a PoC or MVP for the chosen scenario (2 months)
4. Milestone #4: Tested and evaluated/validated the PoC/MVP with members of the target audience (1 month)
5. Milestone #5: Concepts and PoC/MVP refined and elaborated according to feedback and validation results (1 month)
6. Milestone #6: Results reported (all along the internship).

7 Required skills

The candidate must be particularly motivated, have strong software development skills, is familiar with the Eclipse environment and DSLs. Knowledge of HTTP resource APIs (for instance, RESTful ones) OpenAPI and/or AsyncAPI is mandatory.

In addition, a good capacity of abstraction and projection is needed, as this is a very innovative project and the intern is expected to propose solutions to industry problems. Participation in standards through proposals to the OpenAPI committee are possible and would be of high value for the intern as well as for the project itself.

A PhD is possible, depending on the results of the internship.

8 Contact person

Please send your application **only if you meet all the required skills** to Dalila Tamzalit
Dalila.Tamzalit@univ-nantes.fr