

Etienne Pageau

jobs@atnpgo.wtf

<https://atnpgo.wtf/>

Employment History

Oproma Inc.

April 2012 - Present

Software Architect / Senior Developer /
Product Development Team Leader

During my near decade at Oproma I've overseen a wide variety of mostly, but not exclusively, technical tasks resulting in a concentration on full stack development touching all phases of the software development lifecycle. From requirements elicitation to deployment going through application, UI, and UX design as well as implementing said designs. My first couple of years at Oproma were spent on legacy software maintenance followed by a full re-design of the company's existing line of software into a new scalable enterprise-grade headless CMS with web, desktop and mobile client. Annex 1 of this document contains a high level overview of the design decisions that went into the implementation of this suite of applications.

My tasks can be divided into multiple categories. First, let's focus on the server-side aspect of the stack. I've developed both RESTful and non-RESTful APIs alike in Java deployed to tomcat using Java EE 6 Servlet technologies, APIs written in ECMAScript/JavaScript deployed to Node.js using Express.js to handle request routing, and maintenance of legacy ASP.NET Web Forms application deployed to IIS including the implementation of new features integrated via separate APIs communicating over SOAP and WSDL and authentication of users in third-party applications by means of creating a SAML 2.0 identity provider. My more recent implementation of API authentication mechanism was done using OAuth 2.0, including creating the authentication server, integrating the authorization process as part of an existing application, as well as creating new authentication flows for the web and desktop for custom applications and integrating off-the-shelf OAuth plugins for a third-party CMS requiring custom authentication to our services. I've integrated Apache Solr to act as a full-text search engine, including geospatial search, automatically indexing relevant user content and to a lesser extent worked with Apache Lucene, the underlying search library, as well as Apache Tika to extract metadata from files. Acquired significant experience with several traditional RDBMS including but not limited to PostgreSQL, Microsoft SQL Server, and MySQL/MariaDB using both ORM providers, such as Hibernate and Entity Framework, in addition to traditional connectors like JDBC and ODBC ensuring proper care was taken to prevent SQL injection. In addition to these relational data stores, I've also integrated some NoSQL providers such as Apache Cassandra and Redis for various purposes. Porting our existing server stack to a Dockerfile script and authoring a server installation script, with interactive and automated install modes, greatly facilitated new server deployments.

For the application design side of things, I've upgraded existing and created new authentication systems exceeding NIST and OWASP best practices for password hashing and credential management, including two factor authentication using TOTP, together with

implementing secure storage meeting Industry Canada's most stringent published security requirements. Integration of Hashicorp Vault to use Shamir's Secret Sharing to securely distribute server master encryption keys between multiple key stakeholders. I have also spent time building a neural network based text classifier using DL4J as the backing library along with integrating the Inception pre-trained classifiers for automated feature detection in images. Programatically generated, parsed and modified a wide variety of file formats such as PDF, spreadsheets, word processing documents, mailboxes and emails, etc.

On the client aspect of things, I've leveraged web technologies to build applications targeting all the major browsers directly, several desktop environments via Electron, and iOS/Android using first Apache Cordova then React Native. For the purpose of the Oproma software suites, I've written markup directly as well as had it generated using Handlebar templates, React and Vue components as well as using third party libraries to generate sets of .NET controls. Written stylesheets in vanilla CSS and extensively using pre-processors such as SCSS/Sass and Less to reduce development time and complexity by re-using stylings. Worked on both multi-page traditional web forms legacy application with AJAX as well as created new single-page applications, in both cases using a wide variety of client-side libraries and frameworks including but not limited to: Bootstrap, Babel, jQuery, jQuery UI, Mapbox GL, Kendo UI, Telerik ASP.NET AJAX controls, Showdown, Moment, Chart.js, D3, and have developed and published a web-based mobile calendar component. In addition to using web technologies to target all kinds of platforms, I've also used more traditional methods of targeting desktop environment such as using the Java swing toolkit and JavaFX for cross-platform applications and WinForms applications targeting Windows mainly.

Finally, none of this would've been possible without a profusion of non-technical responsibilities and accomplishments such as acting as scrum master overseeing sprint boundary reviews, daily stand-ups and managing Kanban boards. Written a usability testing scenario to be performed by test candidates and reviewed the recording of the testing sessions to improve our application's user experience using human-centred design principles. Wrote unit tests using the JUnit framework to improve the reliability of software builds as well as introduced code linting rules as part of the git commit process to ensure code consistency and readability in order to facilitate future maintenance and indirectly reduce technical debt. Performed post-intrusion forensics work on servers. Brought about continuous integration of our products via Jenkins CI integrated to our internal Gitlab code repositories for traditional products as well as automated documentation generation from our internal wiki documentation. While most comfortable with git, my work has had me use of both Mercurial and Microsoft SourceSafe source repositories. Had to work with most common stacks, such as LAMP, LEMP, LAPP and LEPP on several LTS versions of Ubuntu as well as IIS and SQL Server on Windows Server 2003, 2008, 2012 and 2016, on multiple cloud vendor's platforms such as AWS, Azure, Digital Ocean, Backblaze, Linode, and some OpenStack providers. Lastly, I've also become familiar with a large breadth of softwares such as the Office suite, multiple image manipulation programs, OpenVPN server and multiple clients, emails servers and clients, as well as some level of familiarity with most major cloud vendor's offerings.

Statistics Canada

May 2011 - February 2012

Desktop Support Specialist

My role as a desktop support specialist included troubleshooting hardware and software issues

to triage to various departments. I was also tasked with connecting and setting up various types of peripherals, generally without documentation thus requiring research and documentation of the process for future employees. I was also required to configure various applications, internal and third-party, on user's computers which provided me with large amounts of insight into those various applications

Collège La Cité

Technician

February 2011 – April 2011

My duties as a technician were mainly to assist the students with the use of the various applications installed on the college's informatics lab's computers, troubleshoot the various printer related issues that would arise and perform minor maintenance of those printers. I would also be called on from time to time to provide in-class support to school staff and students as well as overview mass software deployments to a new computer laboratory.

Statistics Canada

Desktop Support Specialist

May 2010 – August 2010

I was hired as part of a group to handle Statistics Canada's migration from Windows XP to Windows Vista. The setup process included scheduling a meeting with the user where we would create a full backup of their data and take inventory of any non-standard software installed on the user's computer. We would then take a new computer, image it with the standard OS image, install the previously inventoried applications while keeping the assets department informed of these and then import the user's data so the computer is ready for use. We would then schedule a second appointment with the user to disconnect their old hardware, connect the new one and return the old one to the inventory department for storage/recycling.

Formal Education

Collège La Cité

Ontario College Advanced Diploma

September 2009 – April 2012

Program aiming to provide an overview of many software and technology fields and the skills required to maintain one's skill set as new technologies emerge. Projects included writing my own very basic but Turing complete programming language targeting a virtual CPU architecture using Lex and Yacc, a network packet sniffer written in C++, various generic CRUD applications using Java, VB.NET and C#, as well as a 2D platformer game built on the now discontinued Microsoft XNA framework. Other classes included, but weren't limited to, multiple networking classes using the CCNA materials including one with a focus on network security. I've also attended IT and domain management classes, electronics laboratories followed by assembly language courses topped off with embedded systems classes where we would develop python applications pushed to development boards over serial. We also participated in multiple mathematics classes including introductions to calculus and statistics as well as a class focused on the skill of implementing arbitrary mathematical equations as code. Our classes and data structures classes gave us deep insight into the Standard Template Library by teaching us how we would go about implementing them from scratch should they not be available. As part of the curriculum, we also received database administration, analysis/system conception, distributed programming, web application development, desktop application interface design, and business management classes.

Annex 1: How to Write Software to Outlive Its Creator

Originally Written August 7th, 2020

Oproma has a very long history with file sharing and collaboration platforms. For this reason, when the time came to reimagine one of our core offerings, many considerations were put into future-proofing the platform, the API and the clients. The goal of this piece is to provide some insights into these considerations.

Being an agile company, the extensibility of the platform is a priority at Oproma. For this reason, we have chosen a heavily modularized architecture for each of the platform's layers: server, web, and mobile; the intent being that any module can be altered or replaced independently.

On the server-side, this manifests itself as clearly defined interfaces that will abstract all operations to be performed on data, whether it be simply storing, indexing, or analyzing this data. A prime example of this is the concept of metadata extractors. In their simplest form, metadata extractors are passed binary data and return a collection of metadata about said data, to be displayed to the end user. Elaborating on the concept, Oproma has also developed extractors with built-in caching for improved performance and, more interestingly, extractors with integrated AI-based image recognition.

A similar approach is taken with our binary data stores, except these are defined as layers within a full data store stack. In its most basic expression, a data store stack will be configured to read and write binary data to a back-end store, whether a locally mounted file system or an object storage service provider out in the universe. Expanding from there, additional layers may be added, according to the various requirements of the specific deployment i.e. an AES-256 encryption layer or a compression layer to save on storage space at the cost of retrieval, etc.

The amalgamation of these ideas results in a server-side API that, with minimal effort from a backend developer, can be deployed in front of any service or back-end provider. This enables front-end developers to write complex, ACL intensive, web applications without having to worry about implementing these themselves while also allowing them to write their code once and be able to deploy it anywhere. Think along the lines of the JVM or hybrid mobile applications but for HTTP APIs. Very little to no prerequisite knowledge of back-end systems is required for front-end developers as integrations are done via standard protocols such as OAuth 2.0.

In the front-end world, the modularity manifests itself in the form of AMD (asynchronous module definitions). Permitting, among other things, a simple and easy way to define and create new folder views; allowing users to consume their data in the way most convenient to them. Furthermore, this modularization allows the application to load modules only as needed. This provides an increased responsiveness, resulting in an improved experience for end users. Another strong advantage of modularized client-side code is the ability to re-use these modules. This was also an advantage for our mobile applications. In this case, after careful evaluation of the options available to us, we embraced Apache Cordova since it meets our current needs in terms of features and performance while also allowing re-use of many of our web modules and styling.

The combination of these platform layers, mixed with intricate physical and environmental security restrictions, results in our most advanced secure file sharing platform ever.