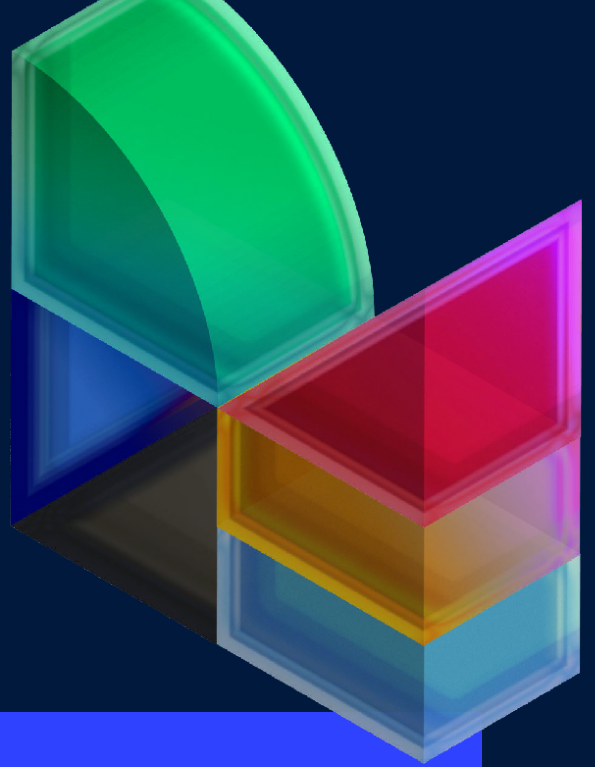


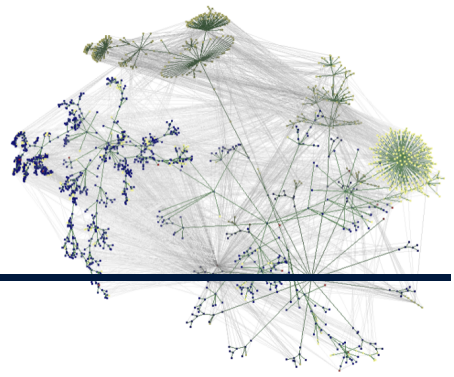


Moderne Platform Technical Overview

WHITE PAPER

The Moderne platform delivers unrivaled automated software modernization at scale. The platform performs source-to-source code transformations for framework migrations, security vulnerability remediations, and code quality issues. This paper provides an overview of the technology behind the Moderne platform and its relationship to the OpenRewrite open-source project.

Moderne as a data warehouse for your code



The Moderne platform is a horizontally scalable data warehouse for your code, where code is imported in a highly structured and dense representation. Queries and transformations can be executed in seconds at scale. In-depth visibility and analysis of your codebase (enhanced with AI) provides insights for action. Moderne can operate on hundreds of millions of lines of code simultaneously.

A foundational technology for the Moderne platform is the Lossless Semantic Tree (LST)—a data model representing the full fidelity of your codebase. This data model is produced out of the source code build in CI or standalone ingest, and then it is stored alongside binaries in your artifact repositories. Once you connect the Moderne platform to your source code repositories, you are able to perform repeatable, automation-driven refactoring and remediation at scale with minimal developer disruption.

An LST is a richer data set than the more common Abstract Syntax Tree (AST). One reason why is that it is *semantic*. The source code that you view in an editor is represented in syntax, but it includes

additional semantic information that is implied by way of linking to an external library or a different type within the project. Syntax-based tools have a hard time grokking all the code dependencies and classes referenced within the code. To illustrate the difference, consider how the same source file would be interpreted with a syntactically aware editor such as Sublime Text. The editor would provide syntax highlighting but would not provide information of the source file compiled in the context of the larger project.

The LST is further evolved from an AST with these characteristics:

- It preserves formatting such as white space and empty classes—even inferring the local and global formatting preferences from observation.
- It preserves the entire type awareness that the compiler possesses in its intermediate representation before producing its final output.
- It is not optimized for grammatical error-tolerance, but it is designed to handle incomplete type information.
- It is designed for round-trip serialization—enabling en masse, fast search and fixes.



Recipes drive expert, automated code transformation

The Moderne platform operates on LSTs using search and transformation recipes from the OpenRewrite ecosystem or your own custom recipes. Basically, recipes are the expert programs that automate code changes.

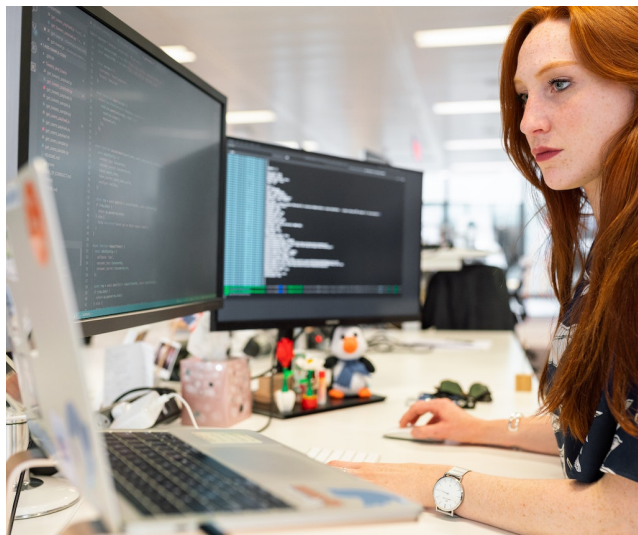
A recipe can represent a single, stand-alone operation, or it can be linked together with other recipes to accomplish a larger goal. This can include everything from simple method renames to a full major framework and dependency migration. Recipes become the single source of knowledge about a code change that can be repeated across repositories.

OpenRewrite is an Apache-licensed environment for creating and configuring the search and transformation recipes. This open source project houses the ever evolving and growing ecosystem of recipes that can be invaluable to independent software vendors, open-source maintainers, and enterprise clients alike.

The Moderne platform enables quick assembly of complex recipes for common code fixes (change method, arguments, package) with a Recipe Builder. It also provides a framework for building recipes for custom searches and transformations. In terms of recipe creation, there is no need for developers to learn a new query language. Recipes are developed in the same language as the code being analyzed and transformed.

The platform automatically adds new recipes to the Recipe Marketplace as they are uploaded from OpenRewrite or created through the Recipe Builder. Your recipes are available across organizations, enabling repeatable, accurate updates everywhere.

A day in the life with Moderne



Moderne operates alongside your software development and delivery processes without imposing any change to your pipelines. Rather, it integrates with your development ecosystem, becoming a developer resource auto-remediating issues and updating code.

While automation replaces manual work, developers are still in control of initiating, reviewing, and putting automated updates through their regular workflows. You can run migration campaigns en-masse across multiple repositories—often performed by a lead migration engineer or architect.

Moderne provides a command line interface (CLI) to publish LSTs out of the build, independent of your CI systems. Once built, the LSTs can be available for platform operations within minutes.

When you run recipes, you are also generating in-depth data tables about your codebase that can be mined for real-time insights and help you focus your automation work. For example, one recipe, called "Find and Fix Vulnerable Dependencies," provides code and library-level software composition analysis (SCA). You are able to see your vulnerabilities, their criticality, the dependency depth, and the patched version (if it exists). The recipe also can update the dependencies automatically.

Developers review automated changes, and issue commits or pull requests to source code in GitHub, GitLab, or Bitbucket. The code fixes are marked as "co-authored by Moderne" for traceability. This way, when your maintenance story completion rate skyrockets, you'll be able to show that these were performed through automation rather than using up developers' valuable time.

Developers can also use the platform to quickly understand their API consumers, as well as what third-party APIs they are using, which databases they interact with, or how to use a specific API.

Moderne is ready for your enterprise

The Moderne platform gives you the flexibility and scale of a SaaS offering with the security of an on-premises installation.

The platform is designed to operate as a private SaaS (a signal-tenant) in your cloud provider (AWS, Google Cloud, or Azure) and region of choice. A Moderne agent on your network connects with your on-prem or cloud-based repositories. If you have multiple isolated environments, multiple agents can be deployed to connect them to the Moderne tenant.

Access is managed through integration with your single sign-on of choice and data encryption is managed through your own encryption key. What's more Moderne provides an audit log and traceability.

And to give you confidence in our platform, Moderne is Service Organization Control 2 (SOC 2®) Type 2 certified. This means that our security processes have met the strict SOC 2 compliance requirements to your data tested through on-site audits of our infrastructure, software, people, procedures, and system data.

We offer a light-weight product evaluation where you can try out the Moderne platform on your own code. We can get set up in your environment quickly and work with you on a pre-defined set of refactoring tasks over the course of a few hours. You'll get to commit real code changes and see immediate value. [Contact Moderne](#) to start your modernization.