

Clause Hackathon

Frequently Asked Questions

Clause Inc

Version 1.0, Feb 18, 2018

Table of Contents

What can I do in the Hackathon?	1
What does Clause do? Why would I use it?	1
Introductory video	2
What is Accord Project?	2
What is a Smart Clause?	3
What is the status of Clause?	3
How do I integrate a Smart Legal Contract with my system?	3
Do you have sample client code I can look at and copy from?	4
How can I create a Smart Legal Contract from existing Smart Clauses?	4
How do I create a new Smart Clause Template?	5
How do I publish my Smart Clause Template so it shows up in Clause?	5
How do I get access to Clause?	5
Do you rate-limit execution requests?	6
What is the Clause pricing model?	6
What does this have to do with blockchain?	6
What connectors do you support?	6
Glossary	8
Index	9



Introduction

*Welcome to the hackathon! **Clause** is here to help you have a great experience, build something amazing, learn and grow! By referring to the answers, tools and samples in this document you should be able to create a Smart Legal Contract and integrate it with existing systems. If you run into difficulties or have questions just ask us for help. We won't bite!*

© 2018 Clause, Inc.

What can I do in the Hackathon?

If we desire respect for the law, we must first make the law respectable.

— Louis D. Brandeis

- ☑ Understand when to use Smart Legal Contracts
- ☑ Create an account on Clause
- ☑ Create a new Smart Legal Contract
- ☑ Execute a Smart Clause in response to an external trigger
- ☑ Create a Smart Clause that performs an external action
- ☑ Create a new Smart Clause Template
- ☑ Add a Smart Clause Template to a Template Library

What does Clause do? Why would I use it?

Clause makes it easy for you to build parameterized legal contracts, parse them to extract machine-readable data, execute them as REST services, and connect them to external data sources.

For example, you may have a Service Level Contract that specifies a percentage reduction in a

customer's monthly bill, based on the availability of a service. The parameters for the contract are defined in a Smart Clause Template and Clause can be used to automatically generate invoices (or move money!) based on IT monitoring events.



Clause is completely generic and can be used to automate the contractual logic behind any business relationship. It makes it easy for you to map from natural language text to executable code.

Smart Legal Contracts based on Smart Clauses solves major challenges for the legal industry, and for commerce generally. Traditional contracts are built by lawyers from natural language templates (often MS Word documents), using relatively unsophisticated find-and-replace of text. The terms in these contracts are then agreed between the contracting parties, the contract is signed, and then archived. In parallel the IT systems are built (or parameterized) to enforce or monitor the terms of the contract. The lack of integration and verification between the IT systems and contract management introduces inefficiencies and often leads to costly errors in contract compliance.

Introductory video



What is Accord Project?

[Accord Project](#) is an open foundation developing techno-legal standards and open-source software tools for the future of smart legal contracting. In particular it is developing the specification and code for Smart Clause Templates and an execution engine under the [Cicero project](#).



Accord Project Cicero is licensed under the Apache-2 license making it easy to integrate with and use within your company.

If you would like to get involved with either the legal or technical side of the Accord Project please visit the [website](#) and sign up! Everyone is welcome and can contribute: ideas, documentation, testing, code, legal templates, models, translations, etc.



Clause Inc. technology is based on and extends the Cicero codebase. If you like analogies, think of the Open Source Git codebase vs. the commercial GitHub service.

What is a Smart Clause?

A Smart Clause is a constituent element of a smart contract. Instead of having to make a full contract executable or 'smart', a user can simply add programmable Smart Clauses to new or existing legal contracts to perform various functionalities. For example, a 'delivery' Smart Clause and a 'payment' Smart Clause may be added to an existing supply contract template. This easily enables legal contracts to be made programmable and executable.

What is the status of Clause?

Clause is currently in closed beta (open to our friends and supporters). Don't worry, if you are attending a Hackathon where we are present we will give you access!

How do I integrate a Smart Legal Contract with my system?

Clause exposes Smart Clauses in contracts as RESTful services, that accept and produce JSON formatted data. Each Smart Clause in a Smart Legal Contract acts as a function, accepting data and returning values. To invoke a Smart Clause you have 3 options:

1. Call a Clause RESTful service, automatically exposed for each Smart Clause in a contract (cross-platform, easiest, requires an Internet connection to Clause)
2. Run a [Cicero Server](#) (cross-platform, can run locally, but resources are managed on the file-system, with limited scalability)
3. Use the [Cicero Engine](#) npm module, and embed execution into your Node.js application (most flexible, but template and clause management must be performed by the application)

We recommend you start with option (1) as there is nothing to install or manage. Simply create a Smart Legal Contract in Clause, and then click the *Connections* tab to see the Execution URL.

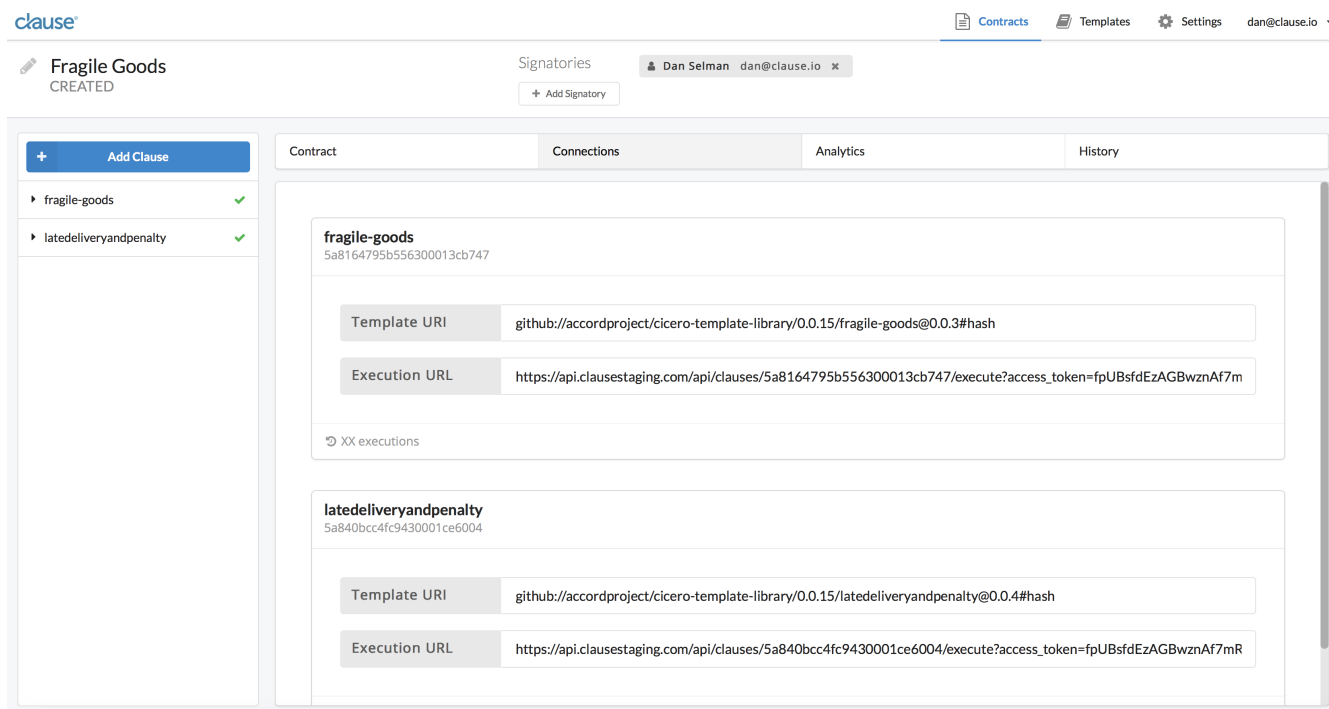


Figure 1. A Viewing Smart Clause Connections

Do you have sample client code I can look at and copy from?

We have published the source code to the [Fragile Goods Demo](#). The demo uses the HTML5 accelerometer API to measure the g force and time taken to deliver a package. When the package is delivered the data is submitted to a Smart Legal Contract to calculate any penalties for late or clumsy package delivery.

The demo illustrates how to:

- ✓ Collect data from a device
- ✓ Call the Execution URL for a Smart Legal Contract running on Clause
- ✓ Display the results of execution in a web user interface

How can I create a Smart Legal Contract from existing Smart Clauses?

To create a new Smart Legal Contract you must:

1. Request a Clause account. The Clause employee attending the Hackathon can help set this up for you.
2. Confirm your email address
3. Login to Clause
4. Press the Templates menu at the top-right

5. Select the first Smart Clause Template that you would like to add to your contract
6. Review the Readme and the Sample Text for the Smart Clause
7. Press the "Create Contract" button to create a new contract that contains the Smart Clause
8. Using the contract editor you can customize your contract, replacing parameters with the values that you would like to use.
9. Save your changes
10. Press the "Connections" tab to view the Execution URL for your Smart Clause
11. Congratulations, you can now invoke your Smart Clause!

How do I create a new Smart Clause Template?

Creating a new Smart Clause Template essentially involves creating a directory structure with some files that conform to the Accord Project Template Specification.

Cicero includes some command-line tools and examples to make creating Smart Clause Templates easier.

If you would like to create your own Smart Clause Template please refer to the Cicero documentation [here](#).

How do I publish my Smart Clause Template so it shows up in Clause?

Smart Clause Templates in Clause are managed in Template Libraries. By default Clause displays the templates in the Open Source Accord Project [template library](#), and the Clause template library. In addition you can add your own private template library to your organization using the Clause user interface.

You are encouraged to fork the [sample template library](#) as the basis for your own template library. You can then add your repository to your Clause account from the template page.



The [sample template library](#) includes further detail on how to configure your template library to make sure that it will appear in Clause.

How do I get access to Clause?

Please come talk to us at the Hackathon, or email support@clause.io.

Do you rate-limit execution requests?

Yes, Clause uses an API Gateway to rate-limit execution requests.



For high-volume/performance scenarios please contact us at support@clause.io.

What is the Clause pricing model?

Clause is currently in closed beta. We have not yet published pricing details, however we expect to use a fairly standard software-as-a-service, pay-as-you-go pricing model.

What does this have to do with blockchain?

Clause integrates with blockchain in a variety of ways, to fulfill different scenarios:

1. Smart Legal Contracts can be invoked from blockchains (passing data from the blockchain into the contract)
2. Smart Legal Contracts can submit transactions to blockchains
3. Smart Legal Contract execution can be embedded in a distributed blockchain node
4. Smart Legal Contract logic can be compiled for execution on the blockchain

(1) is illustrated by the [Cicero Perishable Goods Demo](#), which invokes an out-of-process Cicero Engine (or Clause) from [Hyperledger Composer](#).

(2) is possible using the Clause outbound web connector, which allows contracts to call external services, for example to the [Composer REST Server](#).

(3) is possible for blockchains that support embedded Node.js execution and that can call the Cicero engine, for example [Hyperledger Fabric v1.1](#).

(4) is currently under development.

What connectors do you support?

We are adding connectors at a rapid rate, so this list is not exhaustive! Here is a flavour of the types of things you can do from your Smart Legal Contracts:

Triggers, inbound requests to Clause from the outside world.

- Execute Smart Clauses via authenticated REST services
 - You can form more complex integrations by orchestrating a call to the Clause API with other services using [Node-RED](#) to quickly wire to the IoT, for example.

Actions, cause side-effects in the world with an outbound integration.

- POST data to external REST services (web hooks). [Example](#)
 - This connector can be used to connect to many other online services through [Zapier](#).
 - The [sample](#) provides instructions for Google Sheets, but this can be easily extended to others such as:
 - Clio, for Legal Practice management
 - Xero, for accounting. E.g. your contract could automatically create an invoice.
 - Slack, for notifications. Why not send an alert announcing the outcome of your contract?
- Transfer Ether between Ethereum accounts. [Example](#)

Glossary

Smart Legal Contract

A natural language legal contract that is composed of a set of Smart Clauses.

Smart Clause

Typically a paragraph of natural language text that has precise execution semantics and that can contain parameters. The (Smart Clause has associated executable code (logic) that a computer can execute.

Trigger

An external event or transaction that causes a Smart Contract to be executed.

Action

A Smart Clause can cause actions to occur in external systems via connectors.

Blockchain

Blockchain refers to a type of data structure that enables identifying and tracking transactions digitally and sharing this information across a distributed network of computers.

Index

A

API Gateway, [6](#)
Accord Project, [2](#)
Accord Project Template Specification, [5](#)
Action, [8](#)
action, [1](#)

B

Blockchain, [8](#)
blockchain, [6](#), [6](#)

C

Cicero, [2](#), [5](#), [5](#)
Clause, [2](#), [3](#), [3](#), [4](#), [4](#), [5](#), [6](#), [6](#)
Clio, [7](#)
connectors, [6](#)

E

Ether, [7](#)
Ethereum, [7](#)
Execution URL, [4](#)

G

Google Sheets, [7](#)

I

IoT, [6](#)
integration, [6](#)

J

JSON, [3](#)

P

POST, [7](#)

R

REST, [6](#), [7](#)
RESTful, [3](#), [3](#)

S

Service Level Contract, [1](#)
Slack, [7](#)
Smart Clause, [3](#), [3](#), [5](#), [8](#), [8](#)
Smart Clause Template, [1](#), [2](#), [5](#), [5](#), [5](#)
Smart Legal Contract, [1](#), [1](#), [3](#), [3](#), [4](#), [4](#), [6](#), [8](#)

Smart Legal Contracts, [1](#)

T

Template Libraries, [5](#)
Template Library, [1](#)
Trigger, [8](#)
trigger, [1](#)

X

Xero, [7](#)

Z

Zapier, [7](#)