Decision Framework - Advanced

Preface

- **Purpose**: Exploring COMPOSITE and STP decisions. This document is to be viewed as an extention to Decision Framework - Applications
- **Writer**: Habibollah Hosseinpoor
- **Reader**: Decision makers, who have to document or explain their decision(s).

STP

When all steps of the Decision Framework is automated, you end up with an STP.

P	Stk	S	V	Q	E	J	D	Description
I	T	T	T	T	T	L	X	**Straight-through processing (STP)**: - **templateService**.loadTemplate(templateId) - **externalService**.loadTestResults(templateId,solutions)

```
public DecisionReport decision(String templateId, List<Solution>
solutions) {
    var df = new DecisionFramework(templateId, solutions);
    var template = templateService.loadTemplate(templateId);
    df.setProblem(template.getProblem());
    df.setStakeholders(template.getStakeholders());
    df.setValues(template.getValues());
    df.setQuantifications(template.getQuantifications());
    df.setEvaluators(template.getEvaluators());

    if(solution.isEmpty()){
        df.setSolutions(template.getSolutions());
    } else {
        df.setSolutions(solutions);
    }

    var testResults =
externalService.loadTestResults(templateId,solutions);
    var judgements = MapperService.mapToJudgement(testResults);
    df.setJudgements(judgements);

    var decision = generate(df);
    return decision;
}
```

Examples of STP

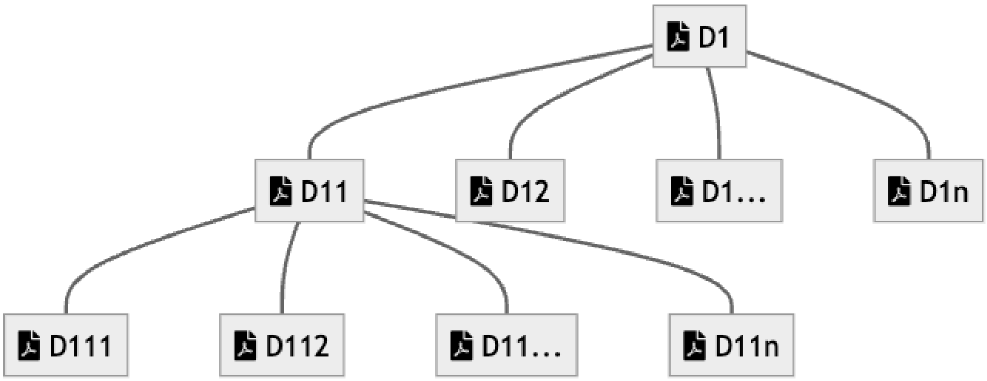
Sample	Prerequisites	Judgement step

Sample	Prerequisites	Judgement step
API test report	First, run your api tests against an API and captures the results in a structured format.	Load test results and map it to judgements in your decision.
Optimization	Given a list of components, with known properties.	Load the known properties of selected components (solutions) and map it to judements

```
# Concrete example of Optimization
P: Create a salad for your friends
Stk: Your friends
V: Health criterias, Nutrients, Minirals, etc...
S: List of row materials
Q: Find out your friends alergies, tastes and preferences and capture it
E: System (STP)
J: Load row materials properties from https://www.matvaretabellen.no/ and
map it
D: Generate decision report
```

Composite Decision

Decision Type	Description
Atomic	Atomic decision can not be broken down into other decisions
Composite	Composite decisions is broken down into multiple decisions



Decision composition: COMPOSITE and ATOMIC

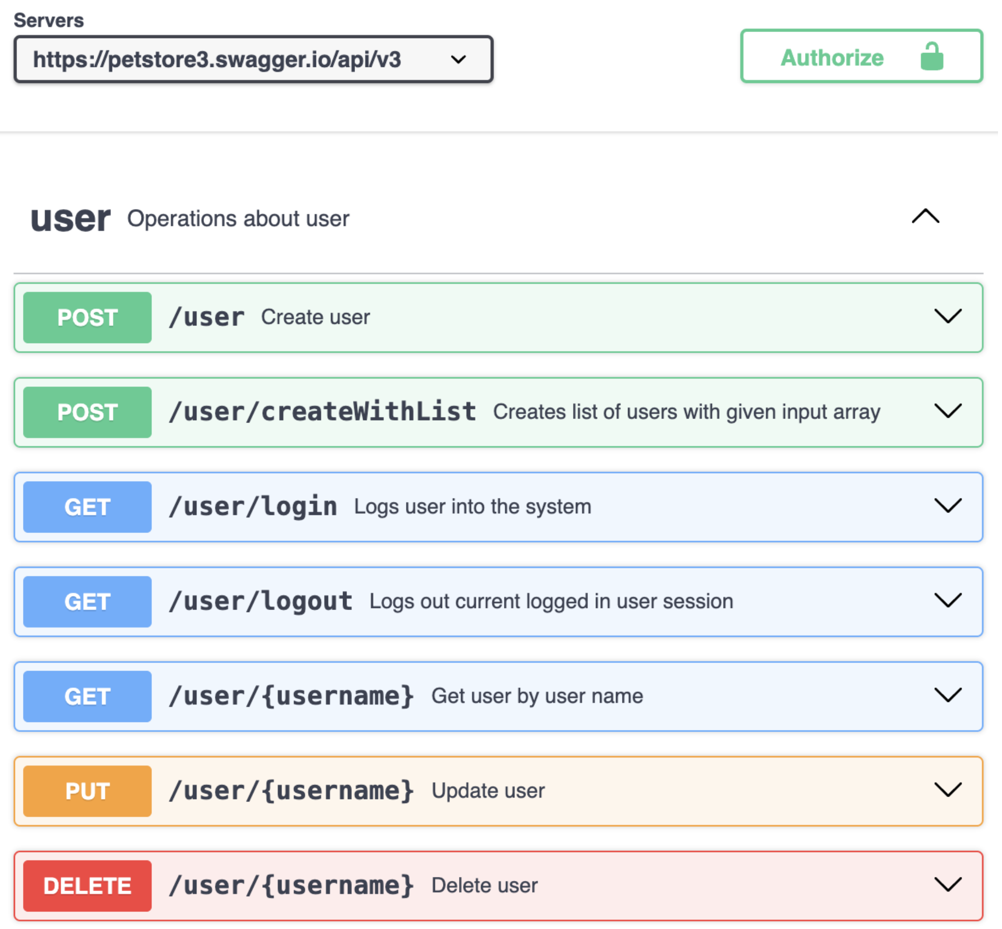
Parent-child relationship for composite decisions

```
- COMPOSITE decision have children
- Both ATOMIC and COMPOSITE decisions can have parent
- Parent.values[i] = Child.problem (child)
- Child.solution = Parent.problem
```

Samples of composite decisions

Sample	Description
OWASP	API Security Testing
ISO 25010	Software Architecture Evaluation

API Security Testing



Servers
https://petstore3.swagger.io/api/v3

user Operations about user

POST /user Create user

POST /user/createWithList Creates list of users with given input array

GET /user/login Logs user into the system

GET /user/logout Logs out current logged in user session

GET /user/{username} Get user by user name

PUT /user/{username} Update user

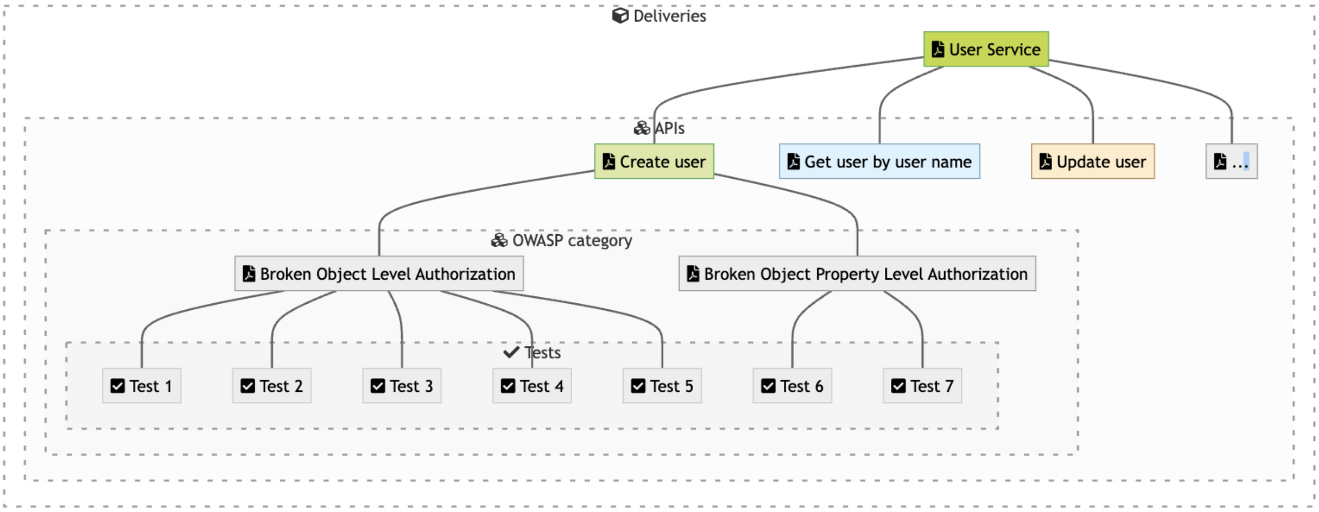
DELETE /user/{username} Delete user

Sample API - petstore, with 7 REST APIs.

Execution

1. Prepare a set of tests for each OWASP API Security top 10 vulnarability.
2. Customize tests for your API(s) (NOT ALL OWASP vulnarability applies to all APIs).
3. Execute the tests againt your API(s), and capture the results.
4. Generate reports.

Deliverables



Deliverables - reports.

Report	Audience	Description	Quantity
Delivery	Product Owner / Security officer	Summary report	1
API	Security officer	API report	7
OWASP	API Developer	OWASP category report. Contains test details (what tests has been performed and the result)	between [20,70]

Total number of reports = between **28 and 78** (1 + 7 + 7 * [2,10]), where all decision reports generation are **100% automated**:

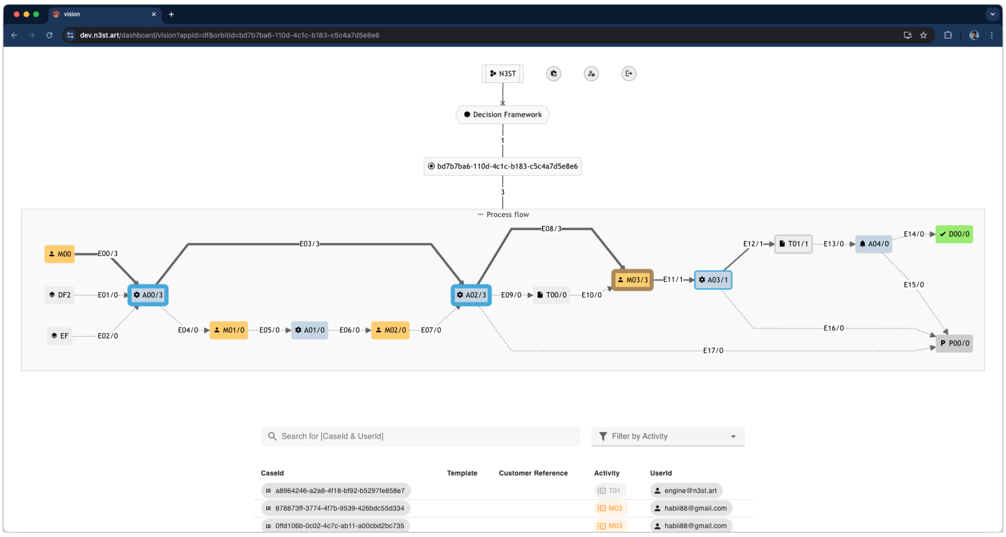
- **Delivery report** : Aggregared (STP), based on API reports
- **API reports** : Aggregared (STP), based on OWASP reports
- **OWASP reports** : Generated (STP), based on test results

Report	Type	Description
API_SECURITY.pdf	Delivery	...
API_SECURITY-A01.pdf	API	...
API_SECURITY-A01-O01.pdf	OWASP	...
API_SECURITY-A01-O03.pdf	OWASP	...
...	...	...
API_SECURITY-A02.pdf	API	...
API_SECURITY-A02-O01.pdf	OWASP	...
API_SECURITY-A02-O03.pdf	OWASP	...
...	...	...

Tools & Methodology

Tools	Description
Decision Framework	Our framework for producing decision reports
N3ST	Our digital printer/factory for production of decision reports
Nodered	API/Test client - To perform actual api invocation and to record the results of the invocations

Software Architecture Evaluation



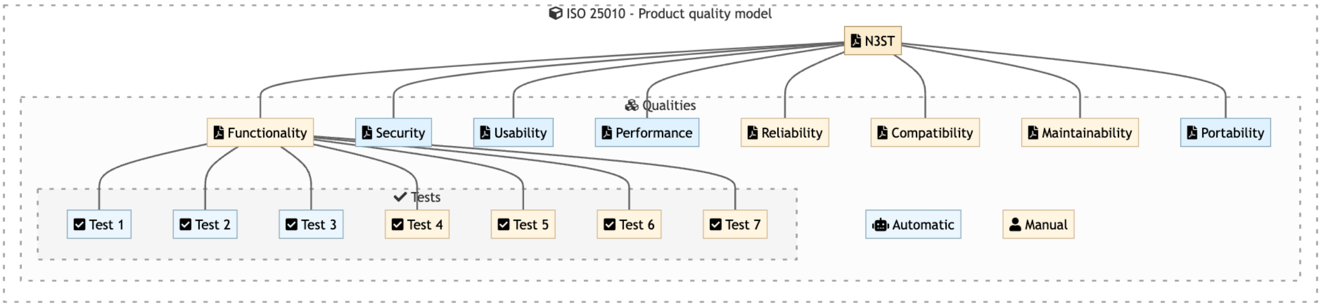
CaseId	Template	Customer Reference	Activity	UserId
a8964246-a2a8-4f18-bf92-b5297fe858e7			T01	engine@n3st.art
878873ff-3774-4f7b-9539-426bdc55d334			M03	habii88@gmail.com
0ffd106b-0c02-4c7c-ab11-a00cbd2bc735			M03	habii88@gmail.com

Sample Software Architecture Evaluation - N3ST.

Execution

1. Prepare a set of tests for each ISO 25010 quality dimension.
2. Customize tests for N3ST.
3. Execute the tests againt N3ST, and capture the results.
4. Generate reports.

Deliverables



Deliverables - reports.

Report	Audience	Description	Quantity
Delivery	CEO	Summary report - product quality evaluation	1

Report	Audience	Description	Quantity
Quality	Product Owner	ISO 25010 quality reports. Contains test details (what tests has been performed and the result)	8

Total number of reports = **9** (1 + 8)

Report generation is **100% automated**, however, not all ISO 25010 quality dimension can be covered with automated tests, thus it must be complemented with manual testing.

- **Delivery report** : Aggregared (STP), based on Quality reports
- **Quality reports** : Aggregared (STP), based on test results

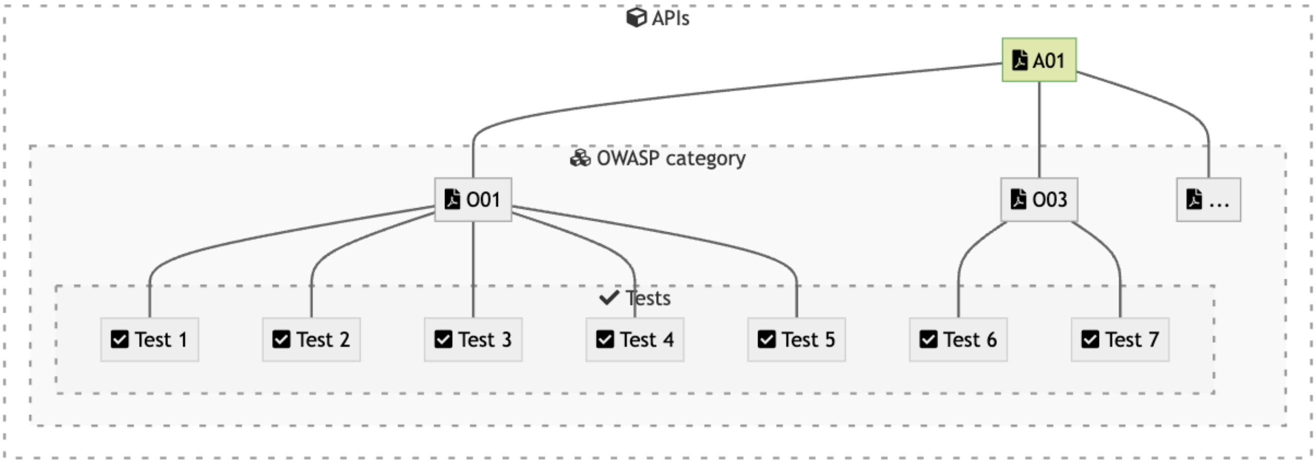
Report	Type	Description
ISO_25010-N3ST.pdf	Delivery	...
ISO_25010-N3ST-Functionality.pdf	Quality	...
ISO_25010-N3ST-Security.pdf	Quality	...
...	...	...

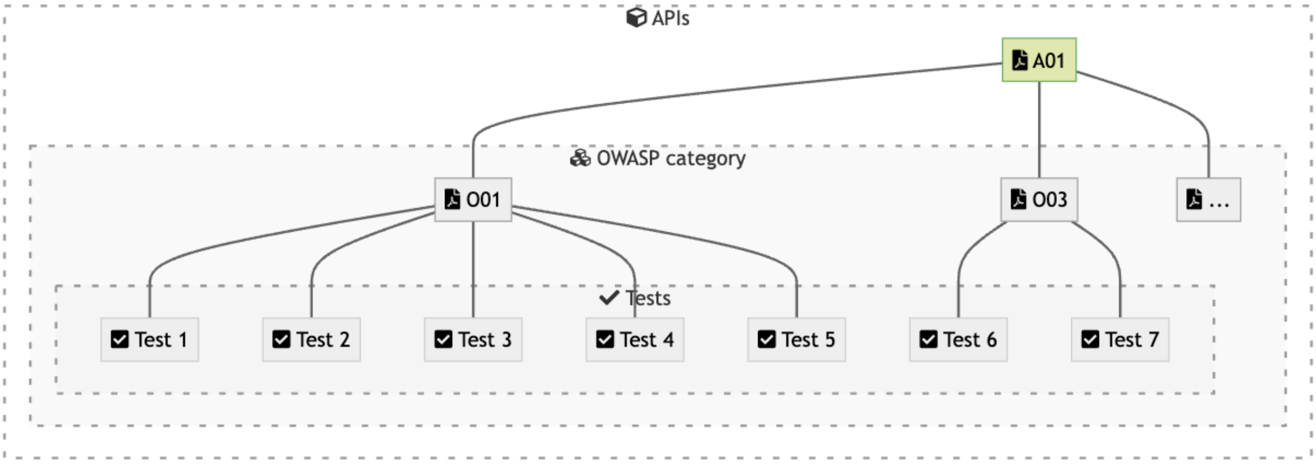
Tools & Methodology

Tools	Description
Decision Framework	Our framework for producing decision reports
N3ST	Our digital printer/factory for production of decision reports
Nodered	API/Test client - To perform actual api invocation and to record the results of the invocations

Changes to Decision Framework

This section shows which changes made to DF for handling Composite and STP decisions, using the following example decision:

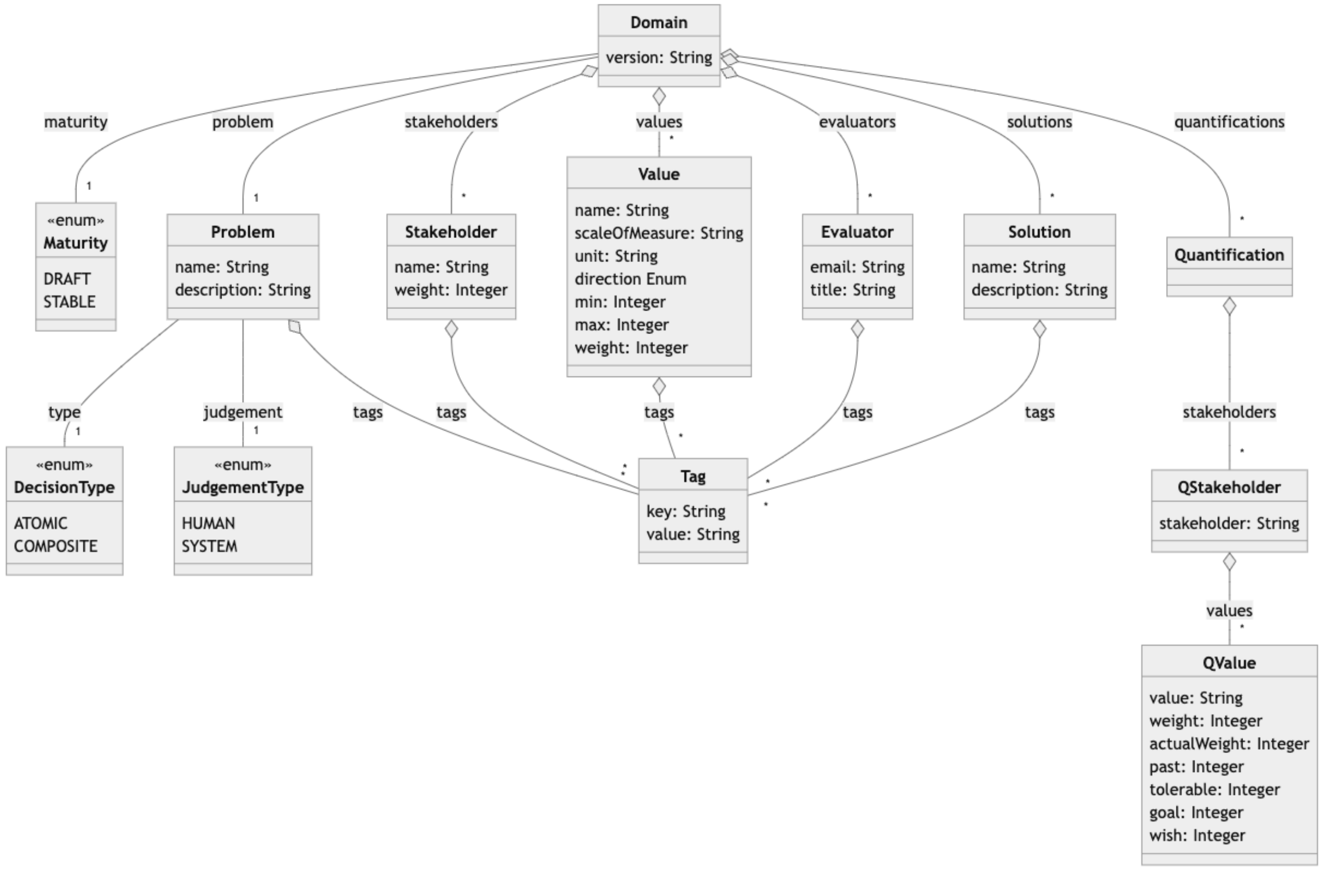




SAMPLE COMPOSITE & STP decision

- **API** (COMPOSITE)
- **OWASP** (STP)
- **Test** test results

Decision Framework Template



DF Template Schema

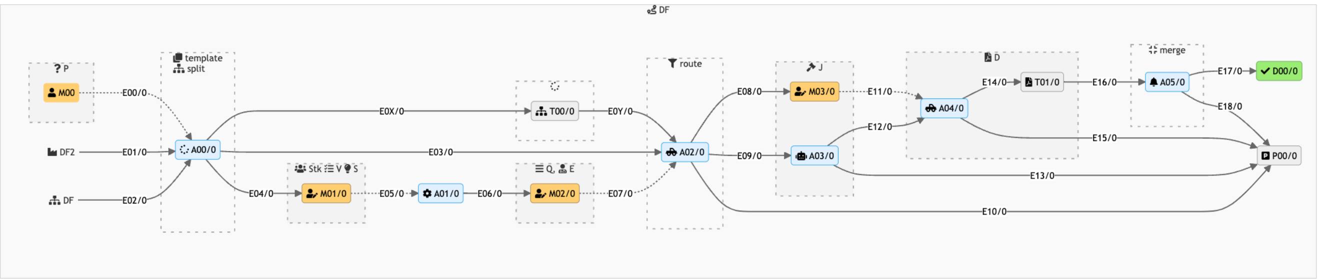
Removed enum:

- **TemplateType** - ATOMIC | SEQUENTIAL

Added enums:

- **DecisionType** - ATOMIC | COMPOSITE
- **JudgementType** - HUMAN | SYSTEM

DF Assembly line



Decision Framework @ N3ST

Functionalities for handling COMPOSITE and STP decisons

Decision	A00	A02	A03	A04	A05
COMPOSITE	- template - split	- initJudgements - route	- load children - aggregate	vsm	- notify parent
ATOMIC STP (Oxx)	- template	- initJudgements - route	- load tests - map	vsm	- notify parent - arkive

`all system (activity) are idempotent`

- **template** : load & apply template
- **split** : create children
- **initJudgements** : calcMachineEvaluations (min, max, tolerable, ...)
- **route** : select human or system judgement. If human judgement, select an eligable judge (caseHandler).
- **aggregate** : **remove a value** (dynamic test results) or **update judgement for a value** for parent
- **notify parent** : update parent that child is processed
- **arkive** : only applicable to cases with DF parent

Next up

Significant goals in the long run:

1. Desision reports to be as simple and as standard/ubiquitous as invoices are
2. An open registry for decisions templates, where everyone can contribute (similar to dockerhub)

If you like our work, and want to contribute to our goals, contact Habib. We ❤️ collaboration.