Graphameleon

Relational Learning and Anomaly Detection on Web Navigation Traces Captured as Knowledge Graphs

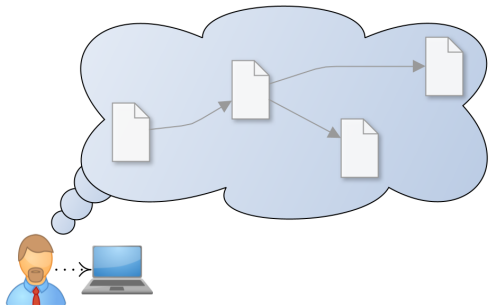
Resource @ TheWebConf 2024

Lionel Tailhardat, Benjamin Stach, Yoan Chabot, **Raphaël Troncy**

Orange & EURECOM

May 13–17, 2024

Context and Motivations: from Web Navigation to Traces Analysis



Scenario Web navigation session

Browsing General search loop scheme

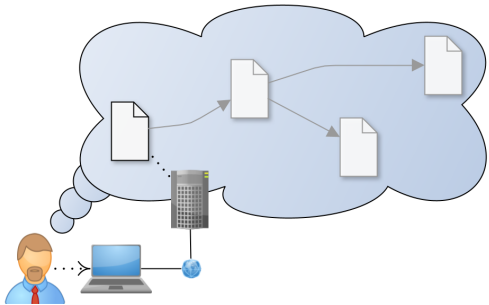
Traces analysis Persona dependent

Web user Green deal, privacy

Platform eng. Resource allocation, performance

Security analyst Malicious activity detection

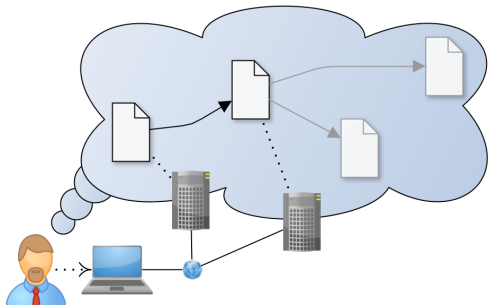
Context and Motivations: from Web Navigation to Traces Analysis



Scenario	Web navigation session
Browsing	General search loop scheme
Traces analysis	Persona dependent

Web user	Green deal, privacy
Platform eng.	Resource allocation, performance
Security analyst	Malicious activity detection

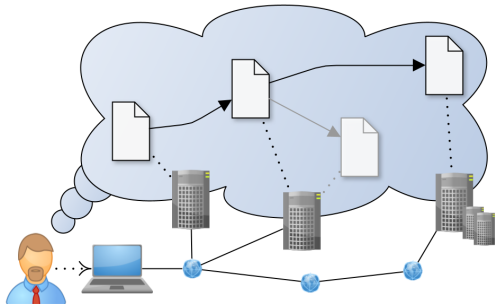
Context and Motivations: from Web Navigation to Traces Analysis



Scenario	Web navigation session
Browsing	General search loop scheme
Traces analysis	Persona dependent

Web user	Green deal, privacy
Platform eng.	Resource allocation, performance
Security analyst	Malicious activity detection

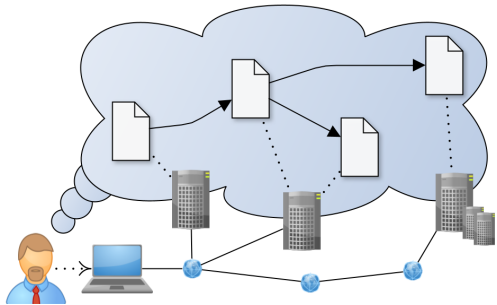
Context and Motivations: from Web Navigation to Traces Analysis



Scenario	Web navigation session
Browsing	General search loop scheme
Traces analysis	Persona dependent

Web user	Green deal, privacy
Platform eng.	Resource allocation, performance
Security analyst	Malicious activity detection

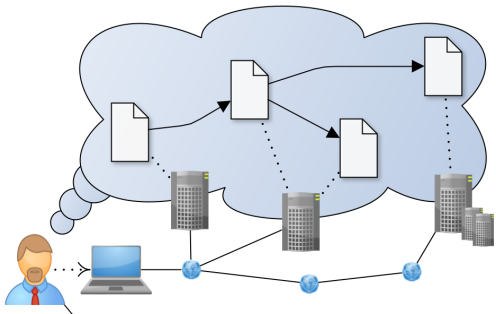
Context and Motivations: from Web Navigation to Traces Analysis



Scenario	Web navigation session
Browsing	General search loop scheme
Traces analysis	Persona dependent

Web user	Green deal, privacy
Platform eng.	Resource allocation, performance
Security analyst	Malicious activity detection

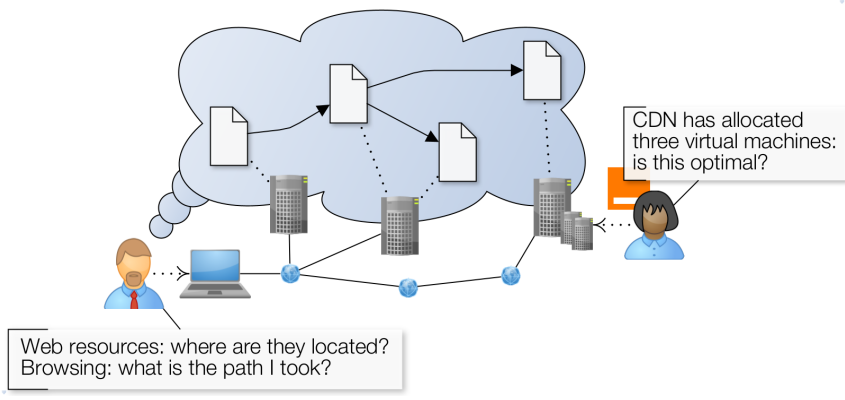
Context and Motivations: from Web Navigation to Traces Analysis



Scenario	Web navigation session
Browsing	General search loop scheme
Traces analysis	Persona dependent

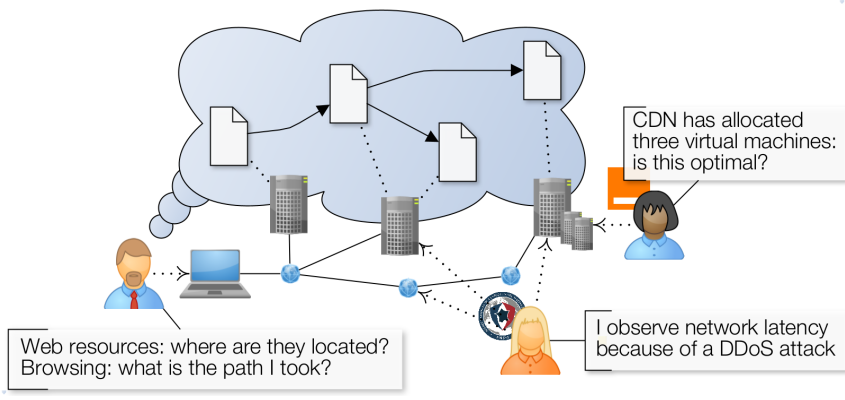
Web user	Green deal, privacy
Platform eng.	Resource allocation, performance
Security analyst	Malicious activity detection

Context and Motivations: from Web Navigation to Traces Analysis



Scenario	Web navigation session	Web user	Green deal, privacy
Browsing	General search loop scheme	Platform eng.	Resource allocation, performance
Traces analysis	Persona dependent	Security analyst	Malicious activity detection

Context and Motivations: from Web Navigation to Traces Analysis



Scenario	Web navigation session
Browsing	General search loop scheme
Traces analysis	Persona dependent

Web user	Green deal, privacy
Platform eng.	Resource allocation, performance
Security analyst	Malicious activity detection

Contextualizing User Actions within the Network Topology?

Web navigation traces What knowledge do traces provide about the structure and dynamics of the network in relation to user activities?

Behavioral model How can we learn a model that takes into account both user activities and network structure?

Challenges

1. Data collection of network and user actions data
   - Encrypted network traffic
   - Private application logs
   - Improper formatting of the data
2. Knowledge representation and reasoning
   - Non obvious cause or purpose of an activity in traces
   - Multiple levels of interpretation of an activity
   - Expert knowledge useful for interpretation stored in third party knowledge bases

Contextualizing User Actions within the Network Topology?

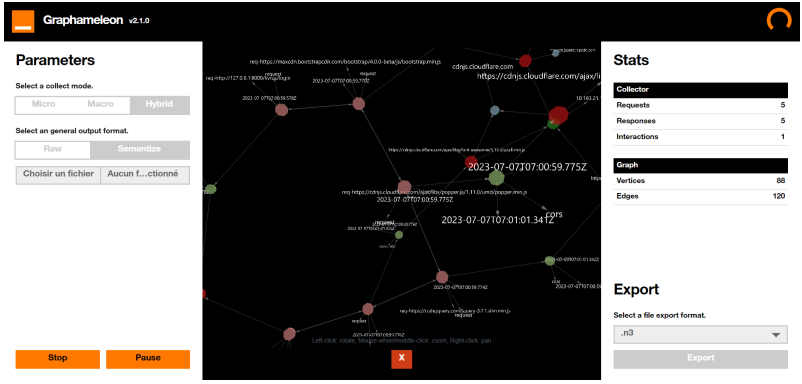
Web navigation traces What knowledge do traces provide about the structure and dynamics of the network in relation to user activities?

Behavioral model How can we learn a model that takes into account both user activities and network structure?

Challenges

1. Data collection of network and user actions data
   - Encrypted network traffic
   - Private application logs
   - Improper formatting of the data
2. Knowledge representation and reasoning
   - Non obvious cause or purpose of an activity in traces
   - Multiple levels of interpretation of an activity
   - Expert knowledge useful for interpretation stored in third party knowledge bases

Experimental Design: the Graphameleon Web extension



Traces Live capture at the browser level of...

- Network requests (macro mode)
- User interactions (micro mode)

Output RDF Knowledge Graph using the UCO ontology

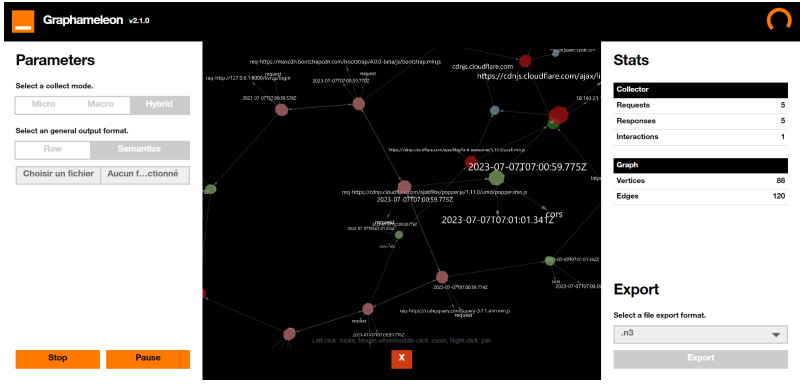
Applications

- Web cartography
- Network behavior analytics
- Anomaly detection

Experiments

- Website complexity clustering
- Navigation trace classification

Experimental Design: the Graphameleon Web extension



Traces Live capture at the browser level of...

- Network requests (macro mode)
- User interactions (micro mode)

Output RDF Knowledge Graph using the UCO ontology

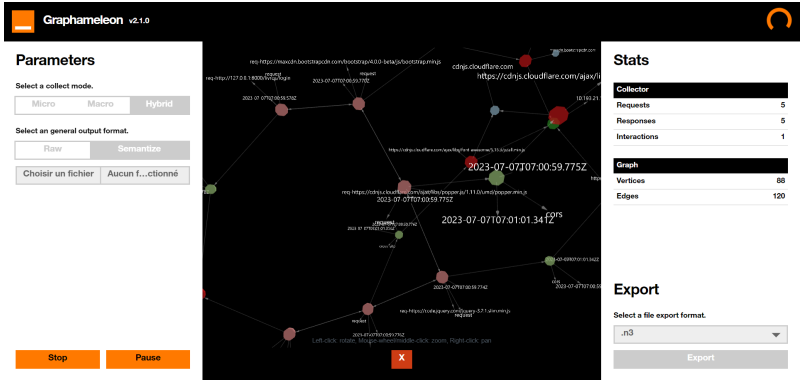
Applications

- Web cartography
- Network behavior analytics
- Anomaly detection

Experiments

- Website complexity clustering
- Navigation trace classification

Experimental Design: the Graphameleon Web extension



Traces Live capture at the browser level of...

- Network requests (macro mode)
- User interactions (micro mode)

Output RDF Knowledge Graph using the UCO ontology

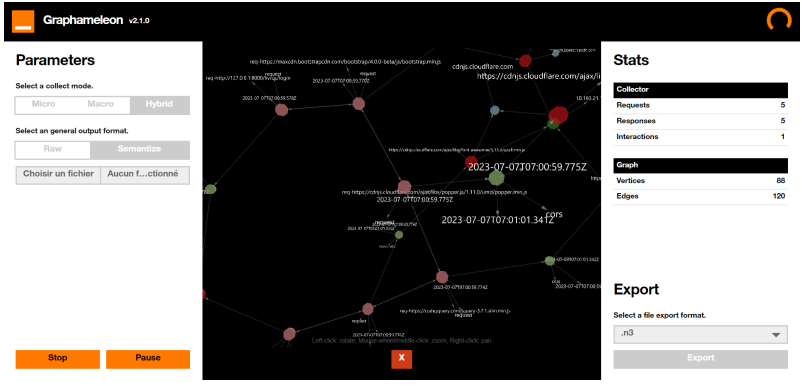
Applications

- Web cartography
- Network behavior analytics
- Anomaly detection

Experiments

- Website complexity clustering
- Navigation trace classification

Experimental Design: the Graphameleon Web extension



Traces Live capture at the browser level of...

- Network requests (macro mode)
- User interactions (micro mode)

Output RDF Knowledge Graph using the UCO ontology

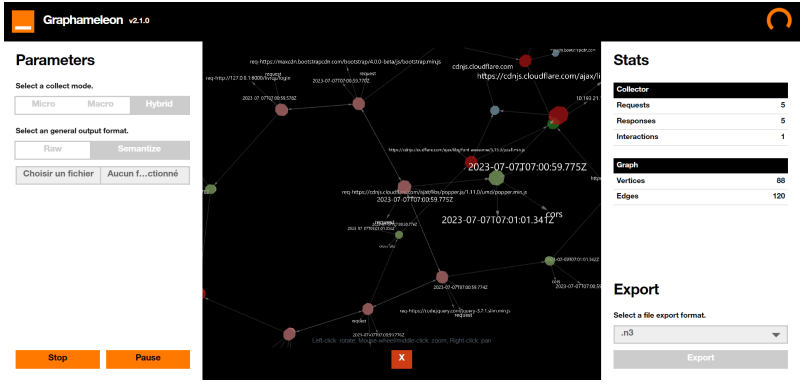
Applications

- Web cartography
- Network behavior analytics
- Anomaly detection

Experiments

- Website complexity clustering
- Navigation trace classification

Experimental Design: the Graphameleon Web extension



Traces Live capture at the browser level of...

- Network requests (macro mode)
- User interactions (micro mode)

Output RDF Knowledge Graph using the UCO ontology

Applications

- Web cartography
- Network behavior analytics
- Anomaly detection

Experiments

- Website complexity clustering
- Navigation trace classification

Evaluation: Website Complexity Clustering

To what extent the behavior of a user visiting a website is crucial in creating a usable footprint subsequently leveraged for anomaly detection?

- Complexity of websites in terms of the number and the size of the resources to be loaded.
- Web navigation sessions with Firefox, anti-tracking $\in \{\text{strict}, \text{standard}\}$, collect mode $\in \{\text{micro}, \text{macro}\}$.
- 27 data collections (three categories $\times$ three sites $\times$ three data collection setups).

	Strict		Standard		Std. / Str.	
	UHC	UIP	UHC	UIP	UHC	UIP
One-Page	61.0	4.0	63.5	7.0	1.04	1.8
Encyclopedia	46.7	6.3	127.7	41.7	**2.73**	**6.6**
Content-Heavy	37.0	6.3	60.7	14.7	1.64	2.3

Average number of entities in micro mode.
Comparison of the average UHC and UIP entities count as a function of the complexity level and of the anti-tracking policy.
UHC = ucobs:HTTPConnectionFacet entities count, UIP = ucobs:IPAddressFacet entities count.

→ User interactions (micro mode): the counts for a given anti-tracking policy configuration exhibit significant variability within each complexity category.

→ Relaxing anti-tracking rules: increase in the average number of connections and remote servers accessed, regardless of the complexity level.

Evaluation: Website Complexity Clustering

To what extent the behavior of a user visiting a website is crucial in creating a usable footprint subsequently leveraged for anomaly detection?

- Complexity of websites in terms of the number and the size of the resources to be loaded.
- Web navigation sessions with Firefox, anti-tracking $\in \{\text{strict}, \text{standard}\}$, collect mode $\in \{\text{micro}, \text{macro}\}$.
- 27 data collections (three categories $\times$ three sites $\times$ three data collection setups).

	Strict		**Standard**		**Std. / Str.**	
	UHC	UIP	UHC	UIP	UHC	UIP
One-Page	61.0	4.0	63.5	7.0	1.04	1.8
Encyclopedia	46.7	6.3	127.7	41.7	**2.73**	**6.6**
Content-Heavy	37.0	6.3	60.7	14.7	1.64	2.3

Average number of entities in micro mode.
Comparison of the average UHC and UIP entities count as a function of the complexity level and of the anti-tracking policy.
UHC = ucobs:HTTPConnectionFacet entities count, UIP = ucobs:IPAddressFacet entities count.

→ User interactions (micro mode): the counts for a given anti-tracking policy configuration exhibit significant variability within each complexity category.

→ Relaxing anti-tracking rules: increase in the average number of connections and remote servers accessed, regardless of the complexity level.

Evaluation: Navigation Trace Classification

Classify Web navigation traces with respect to a procedural model?

- Fitness of captured traces using process discovery $\in$ {Inductive, Alpha, LogSkeleton, Heuristic, AlphaPlus} and conformance checking $\in$ {TokenBasedReplay, Alignment} (PM4Py library).
- Web navigation sessions on a simulated online bookstore website.
- 3 data collection (base scenario with basic authentication, single sign-on authentication, XSS attack scenario).

	Base	Alternative (SSO)	XSS Attack
Request	10	13	11
Interaction	18	14	18
Vertice	263	283	277
Edge	404	431	426

Data collection statistics for the navigation trace classification experiment.

→ User-system usage signature at the knowledge graph level
  - "alternative" = fewer interactions but more network transactions,
  - "XSS attack" = constant interaction count and requests increase by one due to the SQL injection step.

→ Procedural models: help analyzing but lack precision for micro-changes in activities
(98% fitness for the "XSS attack" with the "base" activity model as reference).

Evaluation: Navigation Trace Classification

Classify Web navigation traces with respect to a procedural model?

- Fitness of captured traces using process discovery $\in$ {Inductive, Alpha, LogSkeleton, Heuristic, AlphaPlus} and conformance checking $\in$ {TokenBasedReplay, Alignment} (PM4Py library).
- Web navigation sessions on a simulated online bookstore website.
- 3 data collection (base scenario with basic authentication, single sign-on authentication, XSS attack scenario).

	Base	**Alternative (SSO)**	**XSS Attack**
Request	10	13	11
Interaction	18	14	18
Vertice	263	283	277
Edge	404	431	426

Data collection statistics for the navigation trace classification experiment.

→ User-system usage signature at the knowledge graph level
  - "alternative" = fewer interactions but more network transactions,
  - "XSS attack" = constant interaction count and requests increase by one due to the SQL injection step.

→ Procedural models: help analyzing but lack precision for micro-changes in activities
(98% fitness for the "XSS attack" with the "base" activity model as reference).

Where do I start? Firefox example

1. Download a Graphameleon release, unzip it, and then load it in your Web browser

```
wget https://github.com/Orange-OpenSource/graphameleon/
    releases/download/v2.1.0/graphameleon-dist-2.1.0-firefox.zip
unzip graphameleon-dist-2.1.0-firefox.zip -d graphameleon-ext
firefox about:debugging#/runtime/this-firefox
```

2. Collect, observe and play with your navigation trace data in the Graphameleon Web extension
   - Open the Graphameleon Web extension
   - Select the `Semantize` output format with the `mapping/micro_v2/rules.ttl` mapping rules
   - Click the `Record` button and navigate the Web

3. Export your navigation trace data and analyze it with SPARQL queries
   - Click the `Stop` button and export the trace data as `.ttl` (Turtle syntax)
   - Load your serialized trace data in a SPARQL playground
   - Get statistics on your serialized trace

```
SELECT ?class (COUNT(?subject) as ?classCount)
WHERE { ?subject rdf:type ?class .
FILTER (?class != rdfs:Class && ?class != rdf:Property) }
GROUP BY ?class ORDER BY DESC(?classCount)
```

More in https://github.com/Orange-OpenSource/graphameleon/

Summary and Future Work

The topic Analyze and learn behavioral models of networks (structure and dynamics) in relation to user activities.

Our approach Live capture at the Web browser level of network requests and user interactions + serialization to RDF knowledge graph using the UCO ontology.

Insights User-system usage signature at the knowledge graph level (Web content, tracking policy, attack). Complementary refinements required for leveraging procedural models.

Contribute Contextualizing actions on graph streams? Network structure and behavior databases? ... see the code and just extend it ⇒

Paper

Lionel TAILHARDAT, Benjamin STACH, Yoan CHABOT, and Raphaël TRONCY. 2024.

Graphameleon: Relational Learning and Anomaly Detection on Web Navigation Traces Captured as Knowledge Graphs.

https://doi.org/10.1145/3589335.3651447

Code repository

The Graphameleon Web extension

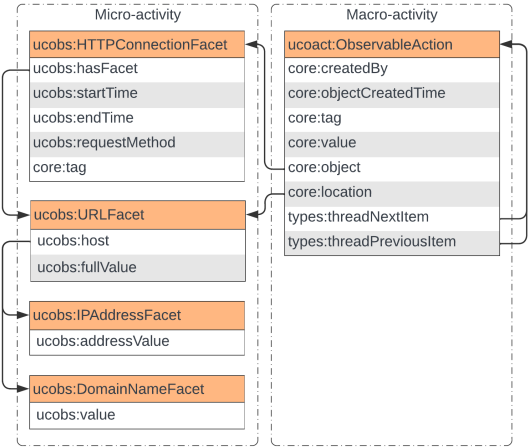
https://github.com/Orange-OpenSource/graphameleon

The Graphameleon dataset

https://github.com/Orange-OpenSource/graphameleon-ds

Additional materials

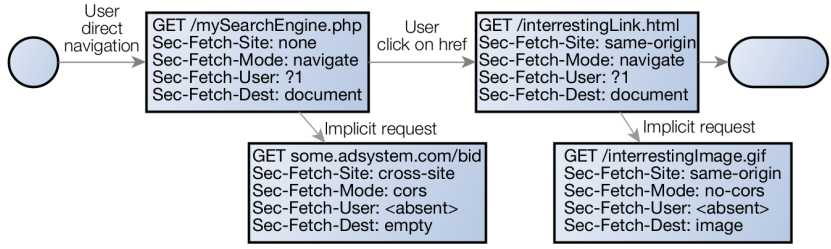
Data model for user activities



- The class diagram defines the concepts and properties used for the semantic representation of micro-activities (left) and macro-activities (right).
- For micro-activities, the presented classes and properties accurately describe a sequence of requests captured at the Web browser level. The `core:tag` property holds the Fetch metadata values (e.g. `core:tag "?1", "document", "navigate", "none";`).
- Macro-activities further enhance the modeling by allowing the description of interactions. The `core:tag` and `core:value` properties hold the user interaction details (e.g. `core:tag "click"; core:value "clc-password-input";`).
- The names of concepts and properties used here are defined within the UCO vocabulary, the following namespaces apply:
  - core = https://ontology.unifiedcyberontology.org/uco/core#
  - ucobs = https://ontology.unifiedcyberontology.org/uco/observable#
  - types = https://ontology.unifiedcyberontology.org/uco/types#

Fetch metadata for inferring the user/equipment activity

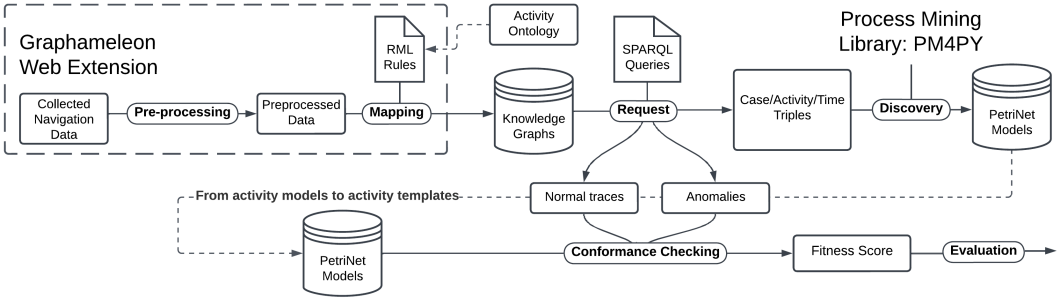
This sequence diagram illustrates the principle for inferring the user/equipment activity from the four fetch metadata HTTP headers through a fictional example of a Web browsing session where the user logs into a search website and follows a hyperlink.



The semantics of fetch metadata summarize as follows:

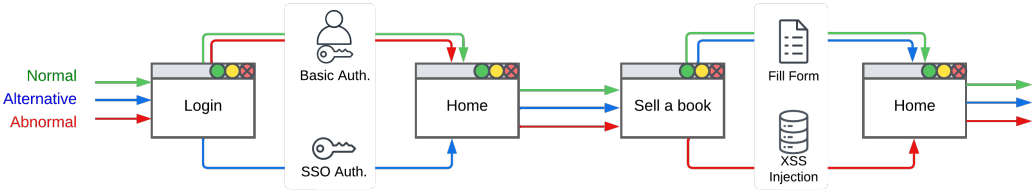
- `Sec-Fetch-Site` = relationship between a request initiator's origin and the origin of the requested resource (e.g. same site, cross site)
- `Sec-Fetch-Mode` = mode of the request (e.g. user navigating between HTML pages vs secondary requests to load images and other resources)
- `Sec-Fetch-User` = only sent for requests initiated by user activation, and its value will always be "`?1`" (e.g. identify whether a navigation request from a document, iframe, etc., was originated by the user)
- `Sec-Fetch-Dest` = where and how the fetched data will be used for better request handling on the server side (e.g. iframe, video component). The sub-documents of each Web page (implicit requests) are identified based on the absence of value for the `Sec-Fetch-Dest` header.

Overview of the Graphameleon data processing pipeline



- The Graphameleon Web extension captures and annotates user activity at the Web browser level.
- A process mining component derives activity models from the resulting RDF KG.
- These models can be used to build a library of activity templates, which are then used by a conformance checking component to classify new activity traces as normal or abnormal activities.

Navigation scenarios for the navigation trace classification experiment



From left to right, the key steps correspond to:

1. User authentication on the simulated online bookstore website,
2. Navigation to the homepage,
3. Purchasing a book,
4. Returning to the homepage.

The arrows correspond to the sequence of steps for each scenario:

- Normal = basic authentication + fill form,
- Alernative = SSO authentication + fill form,
- Abnormal = basic authentication + XSS injection.