



Why and how to start implementing
tag management at the core

The **disclaimer**

Everything is open source



LICENSE

MIT License

Copyright (c) 2022 elbWalker GmbH

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

- ✓ Commercial use
- ✓ Modification
- ✓ Distribution
- ✓ Private use

Feel free to contribute!



alexander.kirtzel



elbwalker/walkerOS

What to **expect**

Things we'll talk about

Analytics implementations are for everyone

Regaining control over data collection

Before the dataLayer pushes

What they all have in common

Origin doesn't matter

The collage consists of five distinct web pages arranged side-by-side:

- Wikia (Screenshot 1):** A dark-themed Wikipedia-like page showing the article "Welcome to Wikipedia". It includes a sidebar with "From today's featured article" and "On this day".
- Reddit (Screenshot 2):** The popular social news and discussion platform. It shows the "Popular" section with various posts and a sidebar for "TOPICS" and "RESOURCES".
- Github (Screenshot 3):** A screenshot of a GitHub repository page for "walker05/walkerOS". It displays code snippets, issues, and pull requests.
- IBM Weather Channel (Screenshot 4):** A weather forecast page for Hamburg, Germany, showing current conditions (32°, Fog), a 24-hour forecast, and a detailed breakdown of snowfall amounts.
- IMDb (Screenshot 5):** The homepage of the Internet Movie Database, featuring the "IMDb BEST OF 2023 DRAMA" banner, movie trailers, and a "Up next" section.

Below the screenshots, there is a vertical column of text that reads:

This is a motherfucking website.
And it's fucking perfect.
Seriously, what the fuck else do you want?
You never build websites and think, your shit is special. You think your 13 megapixel parallax-shit home page is going to get you some fucking Award banner you can glue to the top corner of your site. You think, you don't need jQuery like all \$3 ponytails give it a boner because it finally has bish shadow. Wrong, motherfucker. Let me describe your perfect ass website:

- Shit is lightweight and loads fast
- Fits on all your shitty screens
- Looks the same in all your shitty browsers
- Does not require you to eyeball that visits your site
- Shit is legible and gets your fucking point across (if you had one instead of just 5mb pics of hipsters drinking coffee)

Well guess what, motherfucker:
You. Are. Over-designing. Look at this shit. It's a motherfucking website. Why the fuck do you need to animate a fucking tendy-as-boner flag when I hover over that useless piece of shit? You spent hours on it and added 80 kilobytes to your fucking site, and some motherfucker jobbing it on their iPad with fat sausage fingers will never see that shit. Not to mention blind people will never see that shit, but they don't see any of your shitty shit.
You never knew it, but this is your perfect website. Here's why:
It's fucking lightweight
This entire page weighs less than the gradient-meshed facebook logo on your fucking Wordpress site. Did you seriously load 100kb of jQuery UI just so you could animate the fucking background color of a div? You loaded all 7 fonts of a shitty website just so you could say "ah" at 100px height at the beginning of your site? Your piece of shit.
It's responsive
You dumbass. You thought you needed media queries to be responsive, but no. Responsive means that it responds to whatever motherfucking screen size it's viewed on. This site doesn't care if you're on a iMac or a motherfucking Tamagotchi.
It fucking works
Look at this shit. You've read it - that is, if you can read, motherfucker. It makes sense. It has motherfucking hierarchy. It's using HTML5 tags so you and your bitch-ass browser know what the fuck's in this fucking site. That's semantics, motherfucker.
It has content on the fucking screens. Your site has three bylines and link to your dribbble account, but you spend it over 7 full screens and make me click some bobbing button to show me how cool the jQuery

Some basic principles

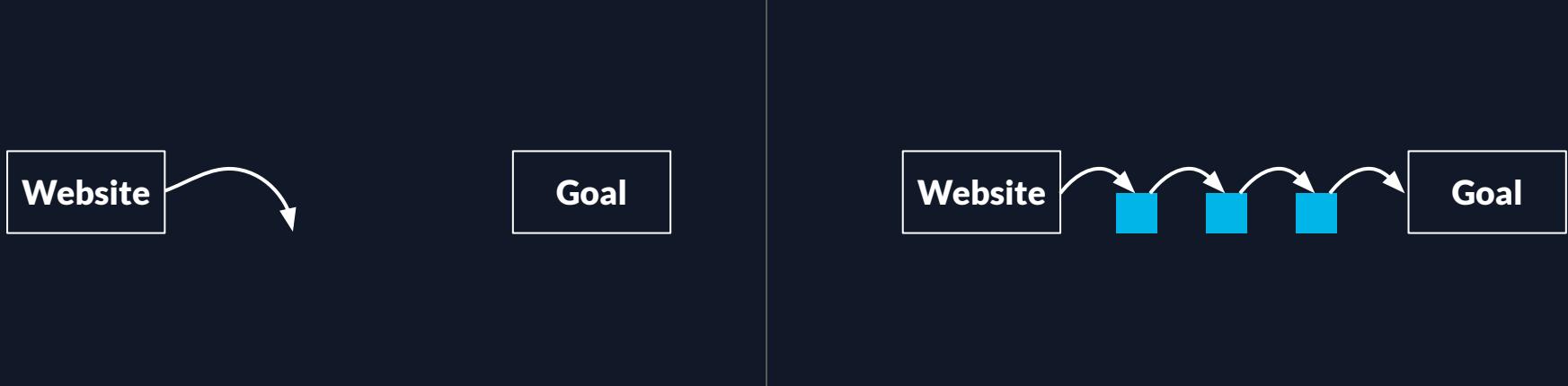
Some basic principles

One piece of wisdom

Step-by-step

Some basic principles

Step-by-step



Some basic principles

One piece of wisdom

**Don't build
on rented land**

Some basic principles

Don't build on rented land



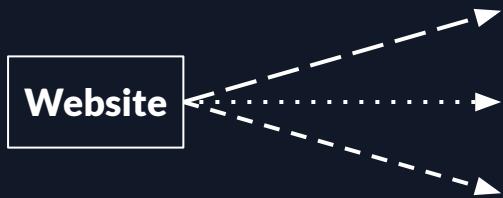
Some basic principles

One piece of wisdom

Resilience
is key

Some basic principles

Resilience is key



Some basic principles

One piece of wisdom

**Focus on what
you can control**

Some basic principles

Focus on what you can control



Google Tag Manager



Google Tag Manager

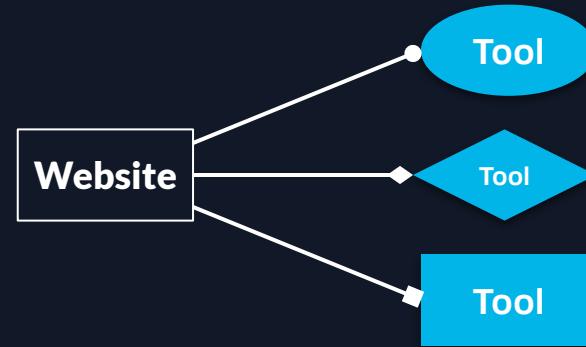
Some basic principles

One piece of wisdom

**Use tools only for
what they were made for**

Some basic principles

Use tools only for what they were made for



Some basic principles

One piece of wisdom

Data is
never done

Some basic principles

Data is never done



Some basic principles

One piece of wisdom

Internalize it

Some basic principles

Internalize it



**Once upon
a time**

Expectations vs. reality

How data analytics has changed

Event —————> Destination

Expectations vs. reality

How data analytics has changed



Data collection **interests**

Different people, different interests, but same origin



Data collection **interests**

Different people, different interests, but same origin



campaign, 3rd party



3rd party

Data collection **interests**

Different people, different interests, but same origin



campaign, 3rd party



consent, namings



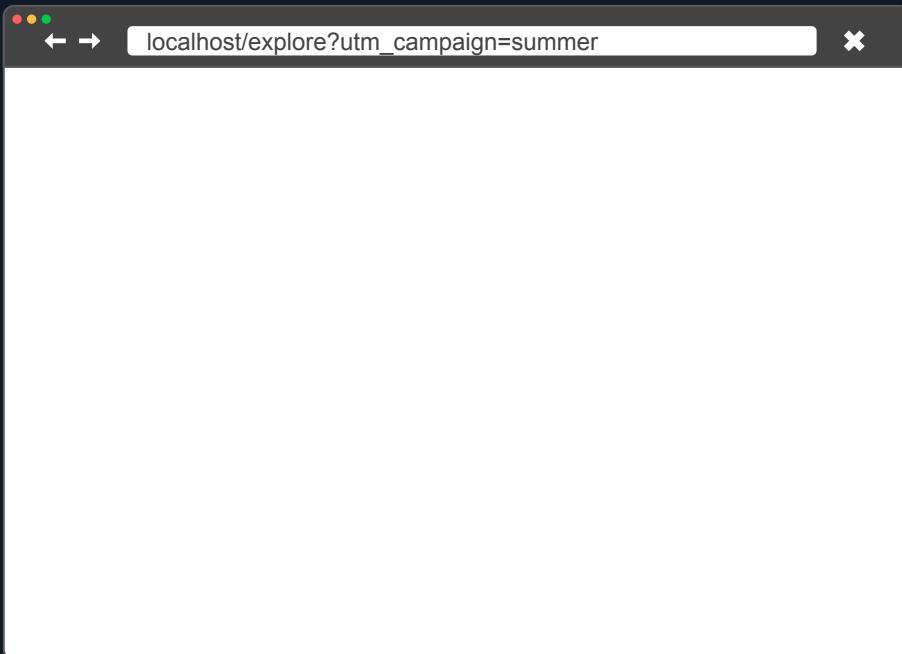
3rd party



consent

Data collection **interests**

Different people, different interests, but same origin



**Online
Marketer**

campaign, 3rd party



**Data
Governance**

consent, namings



**Data
Scientist**

user, config



3rd party



consent



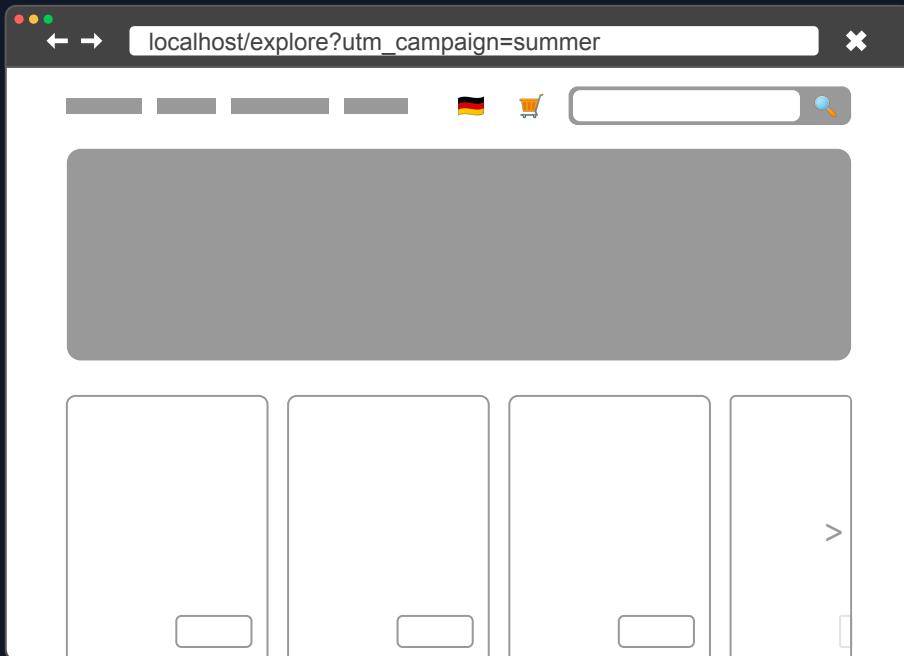
user



config

Data collection **interests**

Different people, different interests, but same origin



**Online
Marketer**

campaign, 3rd party



**Data
Governance**

consent, namings



**Data
Scientist**

user, config



**Product
Owner**

usage, adoption



3rd party



consent



user



config

Data collection **interests**

Different people, different interests, but same origin



**Online
Marketer**

campaign, 3rd party



**Data
Governance**

consent, namings



**Data
Scientist**

user, config



**Product
Owner**

usage, adoption



**Data
Analyst**

events, content



3rd party



consent



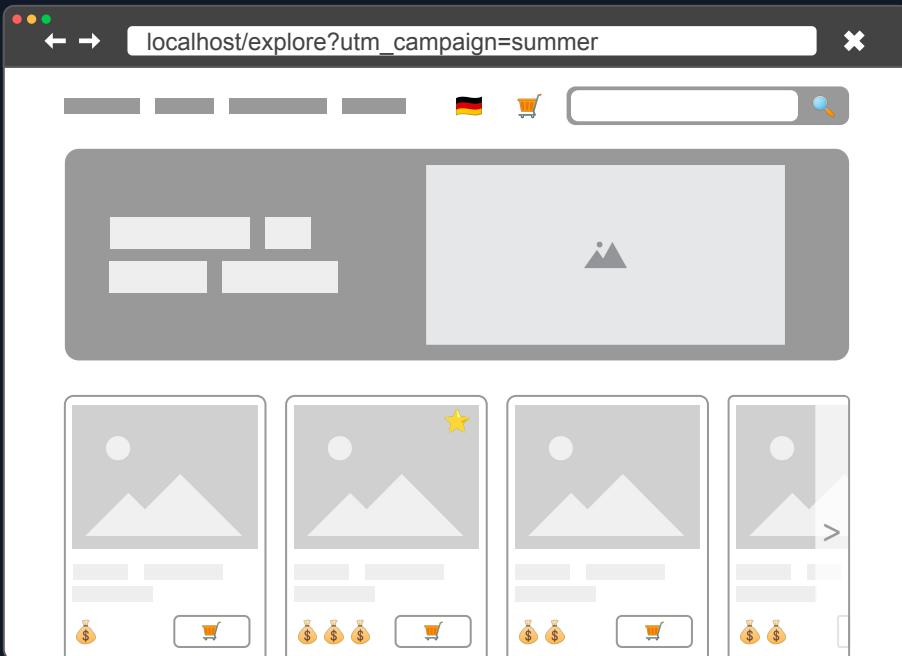
user



config

Data collection **interests**

Different people, different interests, but same origin



3rd party



consent



user



config



**Online
Marketer**

campaign, 3rd party



**Data
Governance**

consent, namings



**Data
Scientist**

user, config



**Product
Owner**

usage, adoption



**Data
Analyst**

events, content

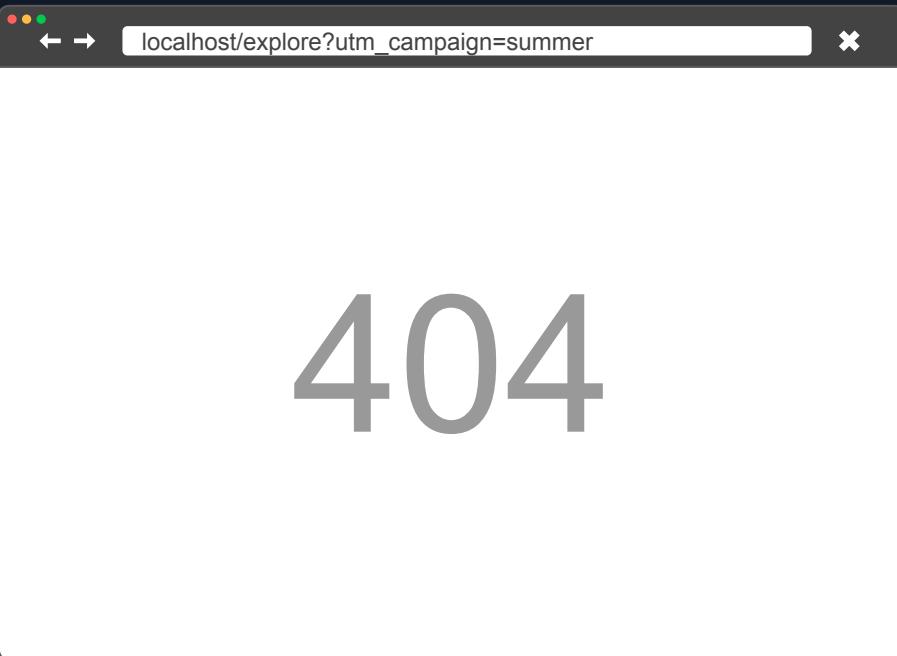


**Growth
Manager**

experiments, behaviour

Data collection **interests**

Different people, different interests, but same origin



3rd party



consent



user



config



**Online
Marketer**

campaign, 3rd party



**Data
Governance**

consent, namings



**Data
Scientist**

user, config



**Product
Owner**

usage, adoption



**Data
Analyst**

events, content



**Growth
Manager**

experiments, behaviour

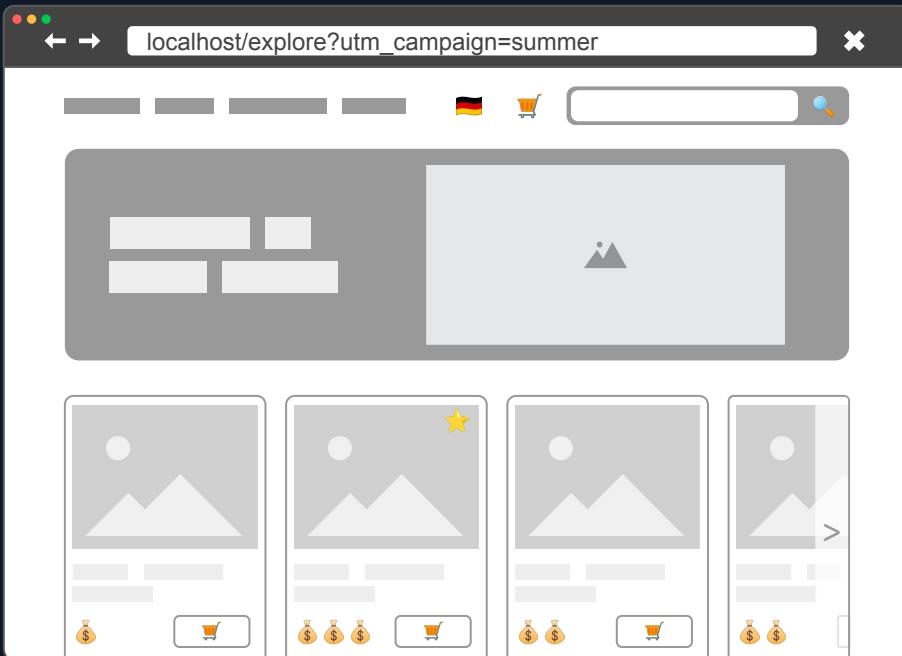


**DevOps &
Developer**

peace of mind

Data collection **interests**

Different people, different interests, but same origin



3rd party



consent



user



config



**Online
Marketer**

campaign, 3rd party



**Data
Governance**

consent, namings



**Data
Scientist**

user, config



**Product
Owner**

usage, adoption



**Data
Analyst**

events, content



**Growth
Manager**

experiments, behaviour

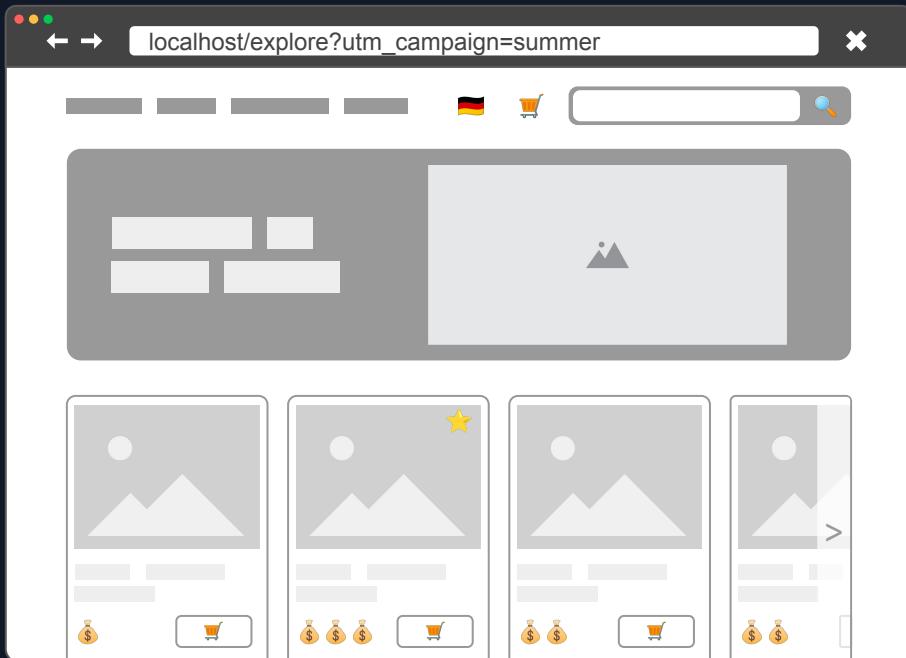


**DevOps &
Developer**

peace of mind

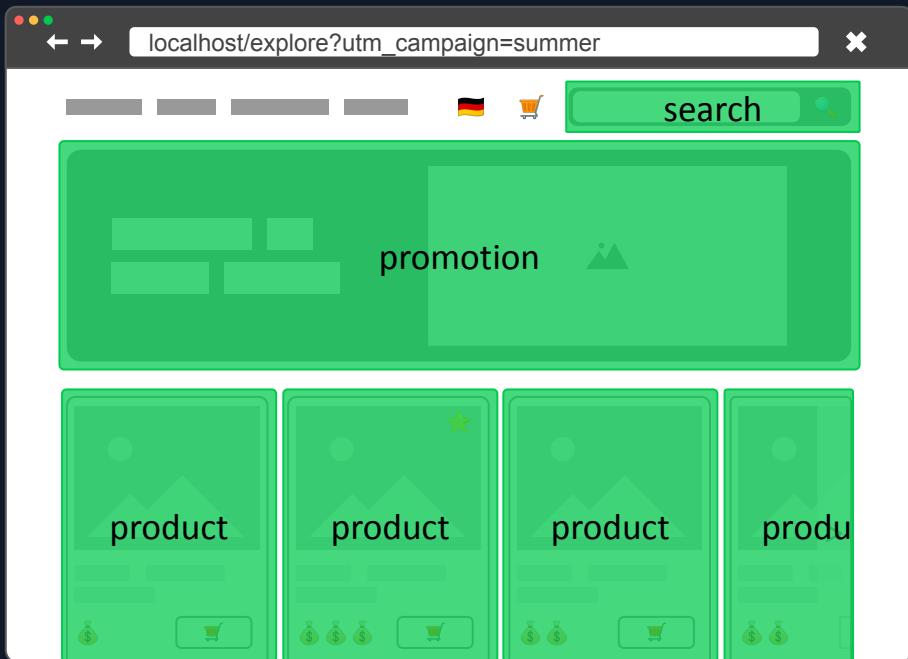
Generic event structure

How to describe an event completely



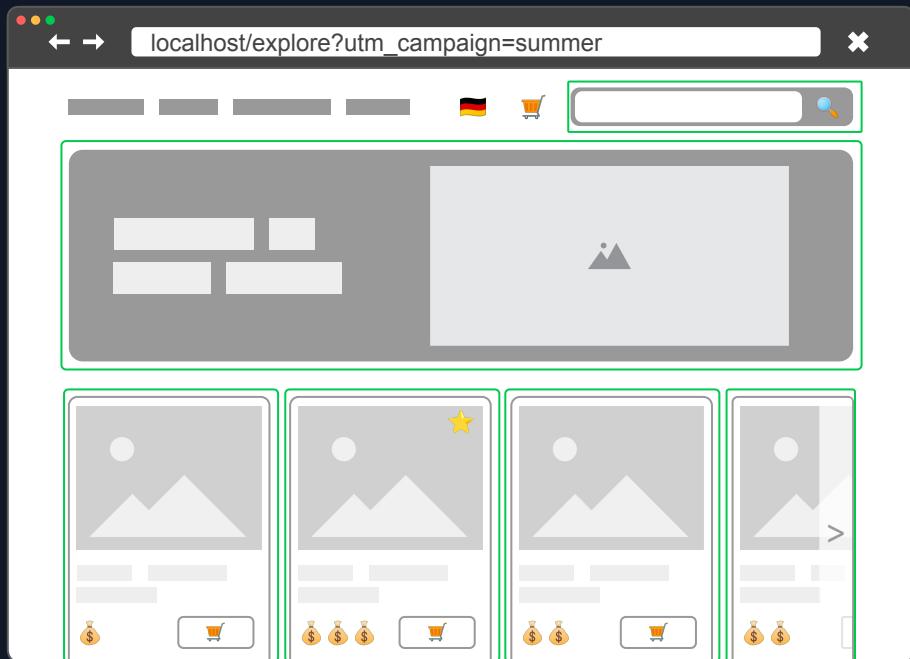
Generic event structure

How to describe an event completely



Generic event structure

How to describe an event completely

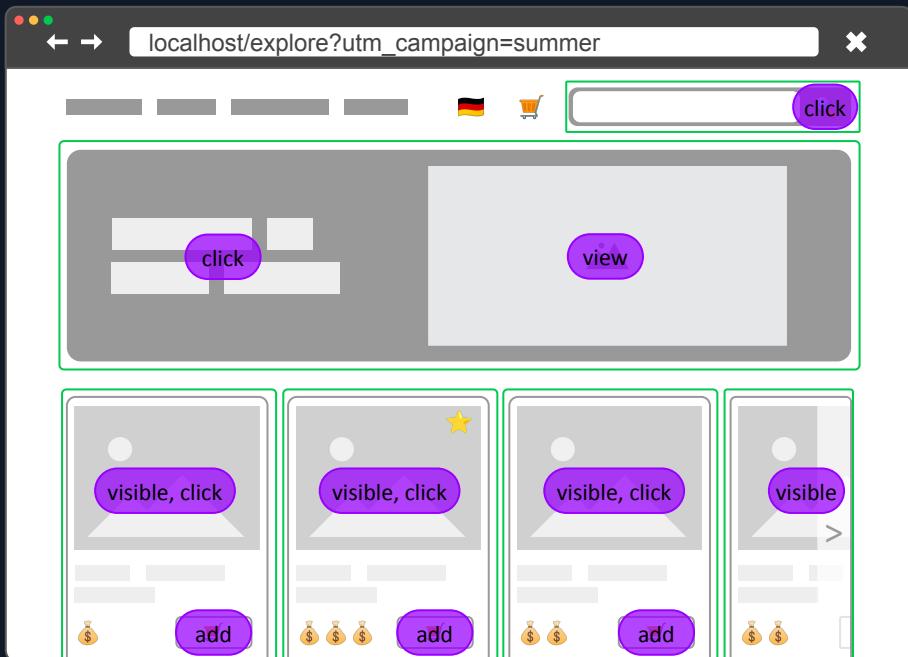


promotion

Entity

Generic event structure

How to describe an event completely

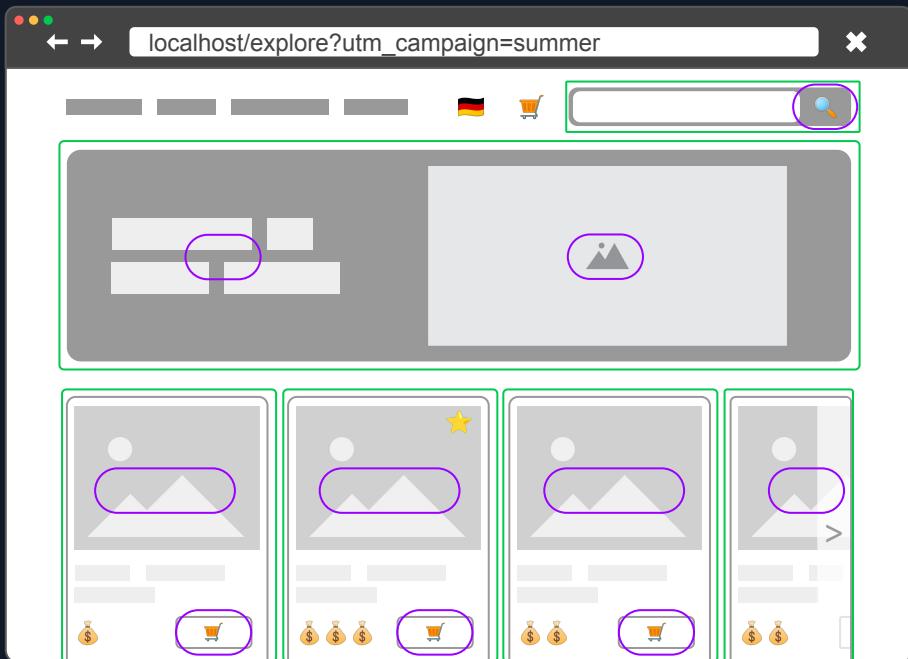


promotion

Entity

Generic event structure

How to describe an event completely



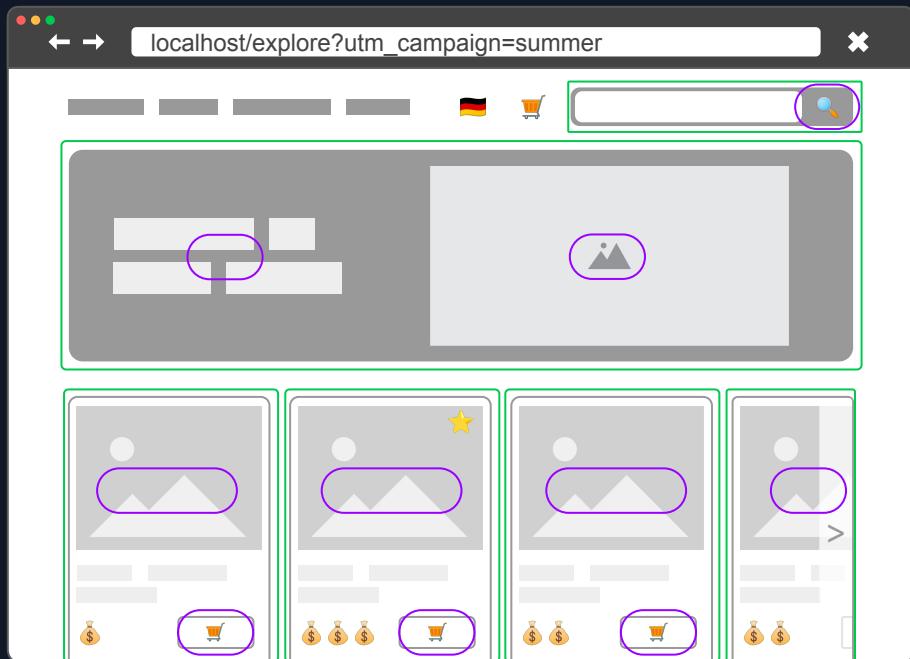
promotion view

Entity

Action

Generic event structure

How to describe an event completely



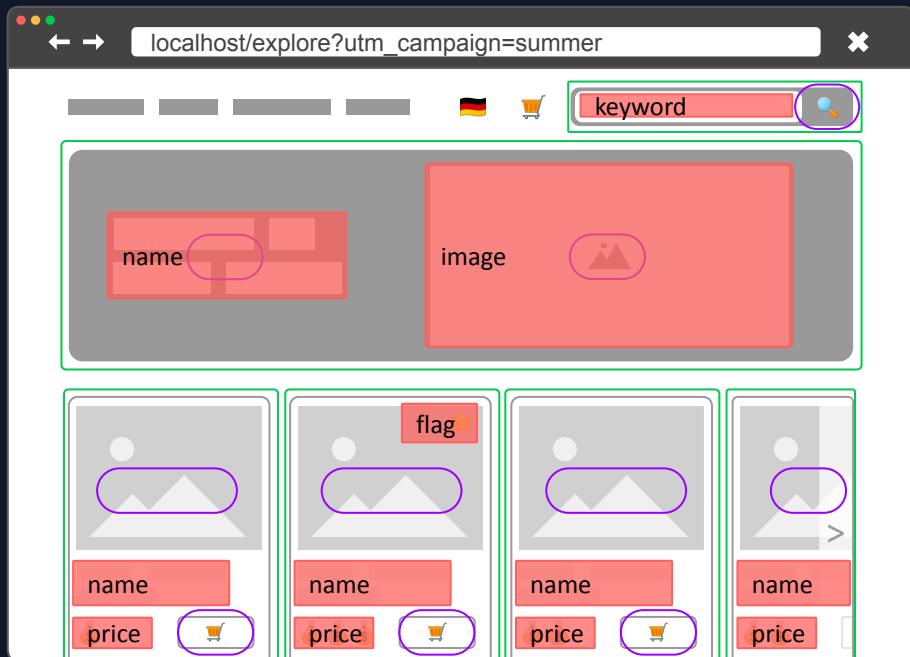
event: promotion view

Entity

Action

Generic event structure

How to describe an event completely



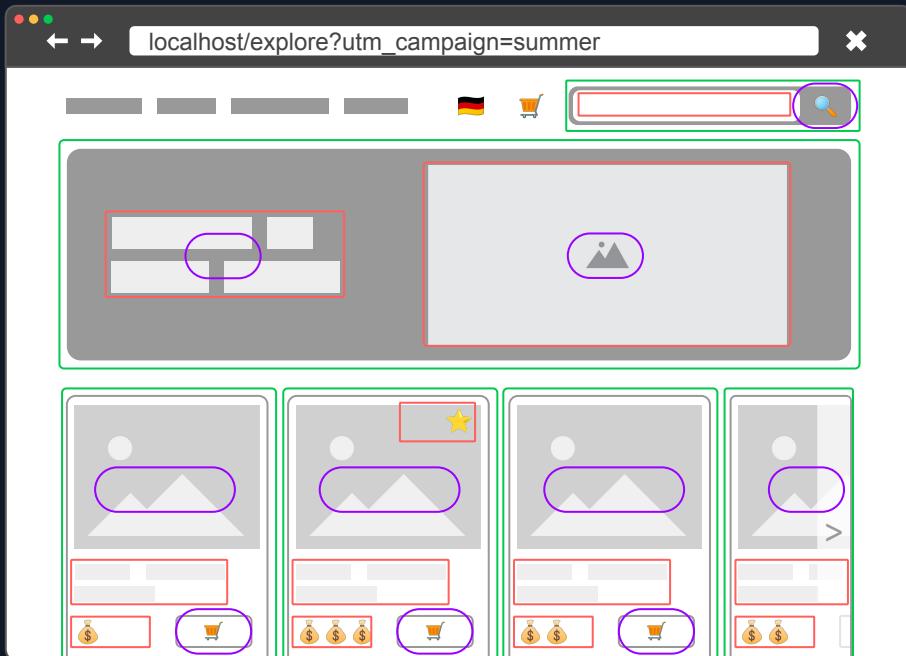
event: promotion view

Entity

Action

Generic event structure

How to describe an event completely



```
event: promotion view
```

```
data:
```

```
name: How to tag
```

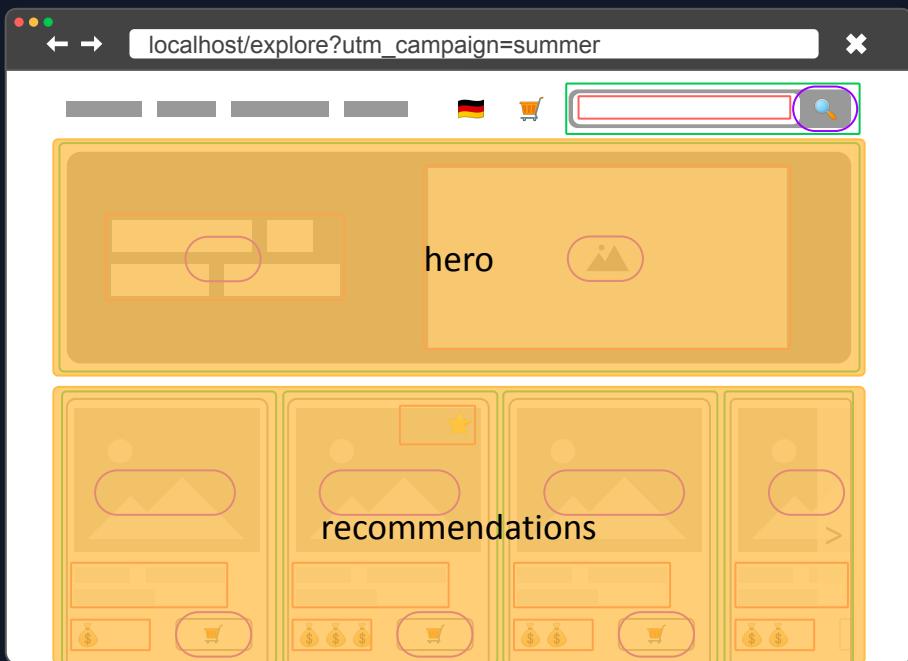
Entity

Action

Data

Generic event structure

How to describe an event completely



```
event: promotion view
```

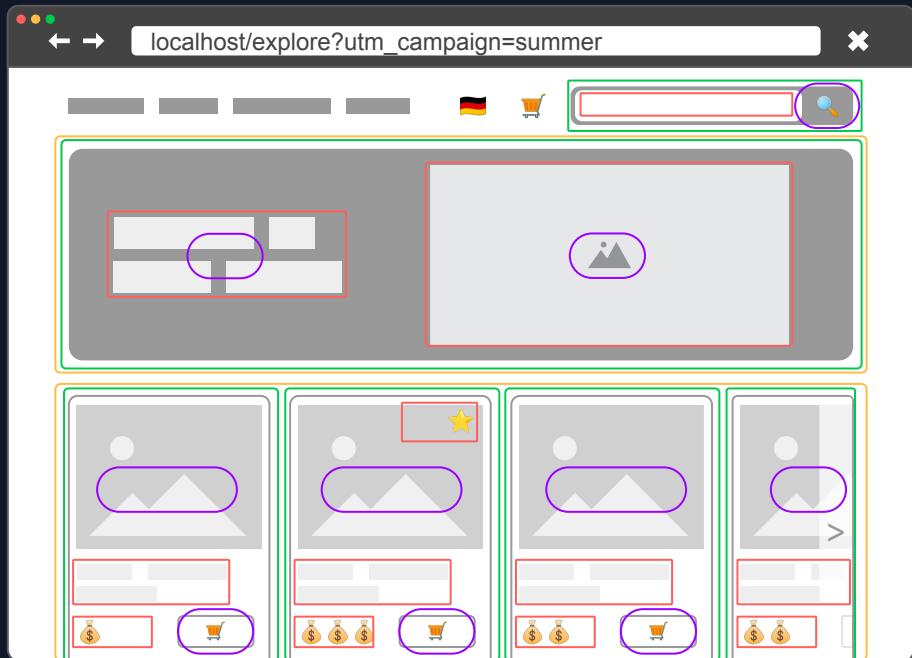
```
data:
```

```
  name: How to tag
```

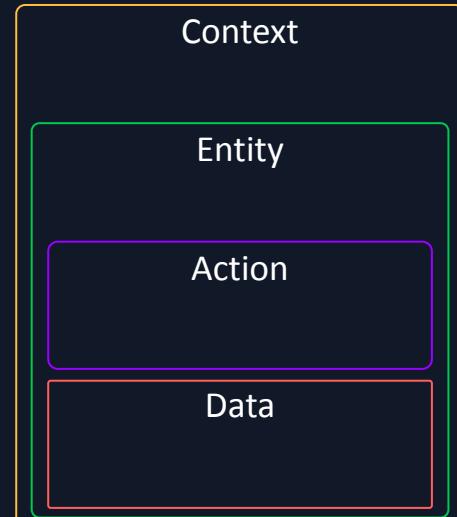


Generic event structure

How to describe an event completely

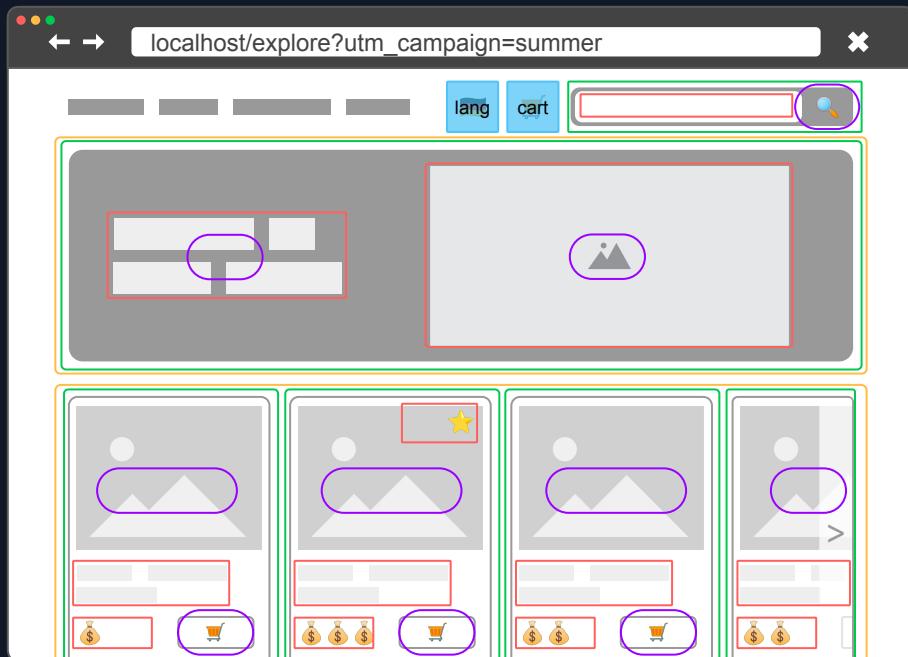


```
context:  
position: hero  
  
event: promotion view  
  
data:  
name: How to tag
```



Generic event structure

How to describe an event completely



context:

position: hero

event: promotion view

data:

name: How to tag

Context

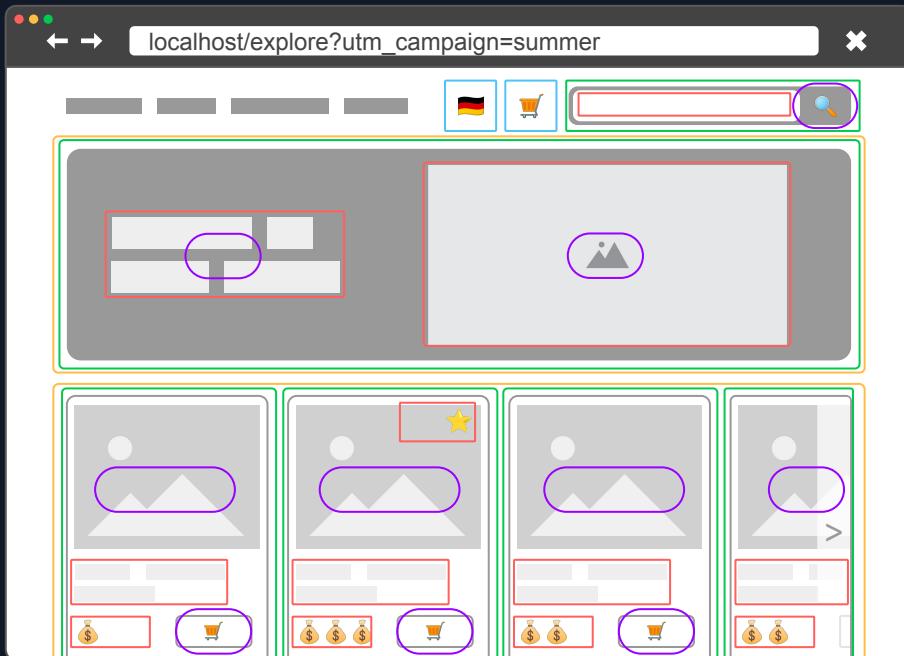
Entity

Action

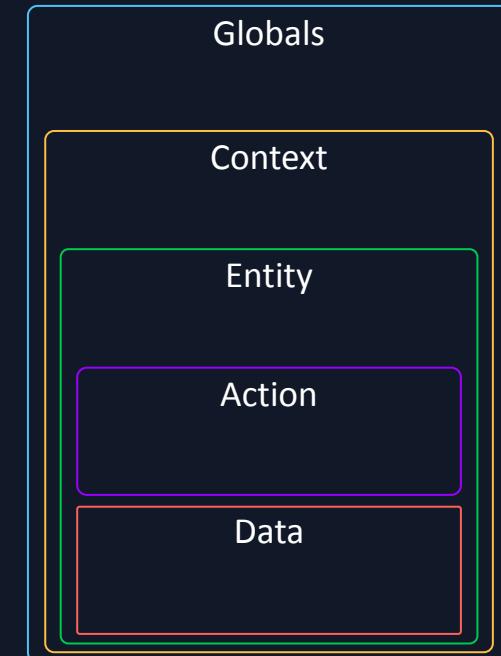
Data

Generic event structure

How to describe an event completely

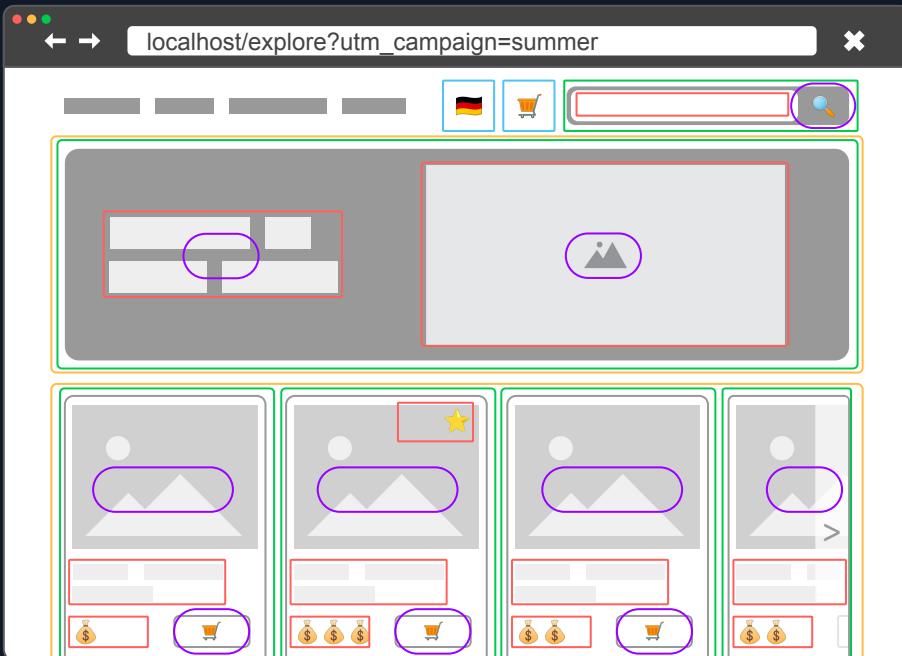


```
globals:  
  language: en  
  
context:  
  position: hero  
  
event: promotion view  
  
data:  
  name: How to tag
```

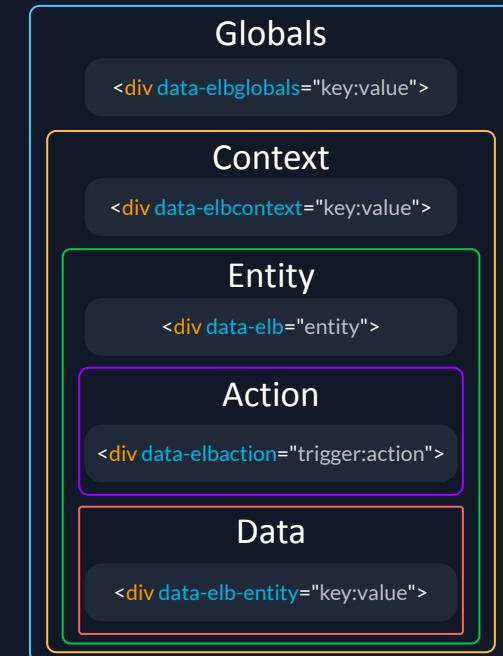


Generic event structure

How to describe an event completely

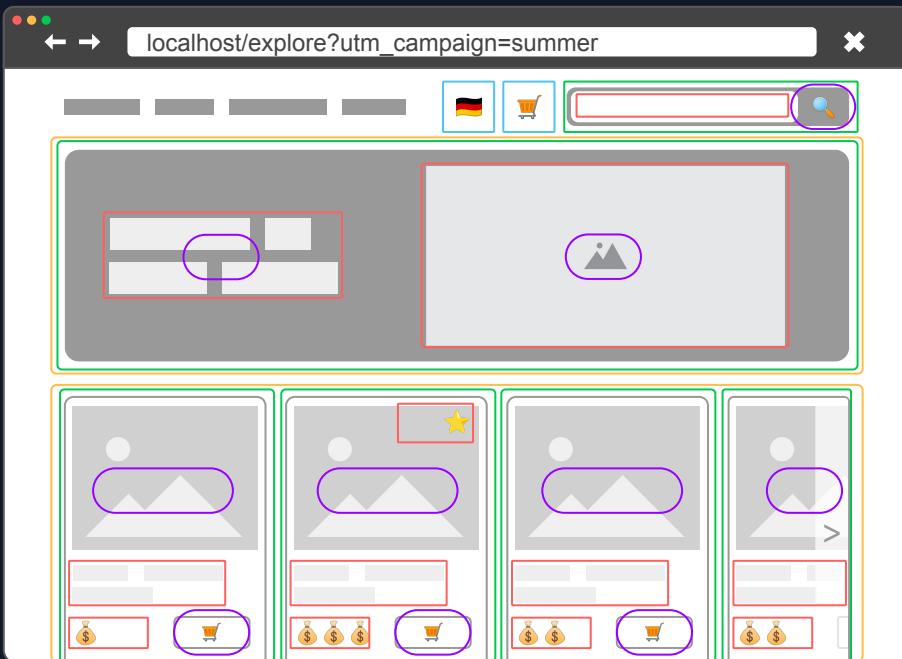


```
globals:  
  language: en  
  
context:  
  position: hero  
  
event: promotion view  
  
data:  
  name: How to tag
```

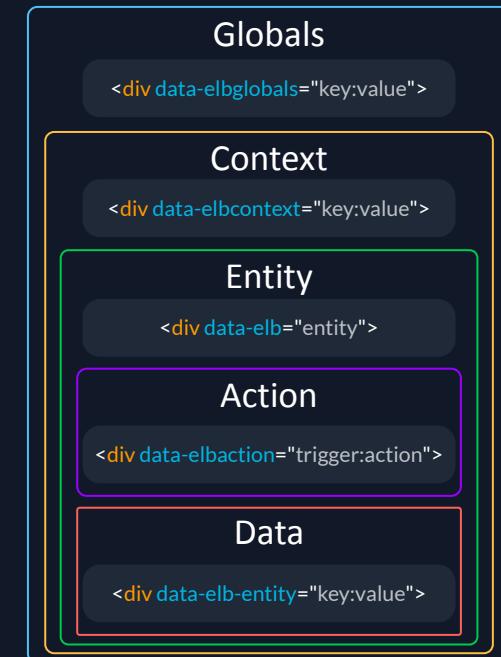


Generic event structure

How to describe an event completely

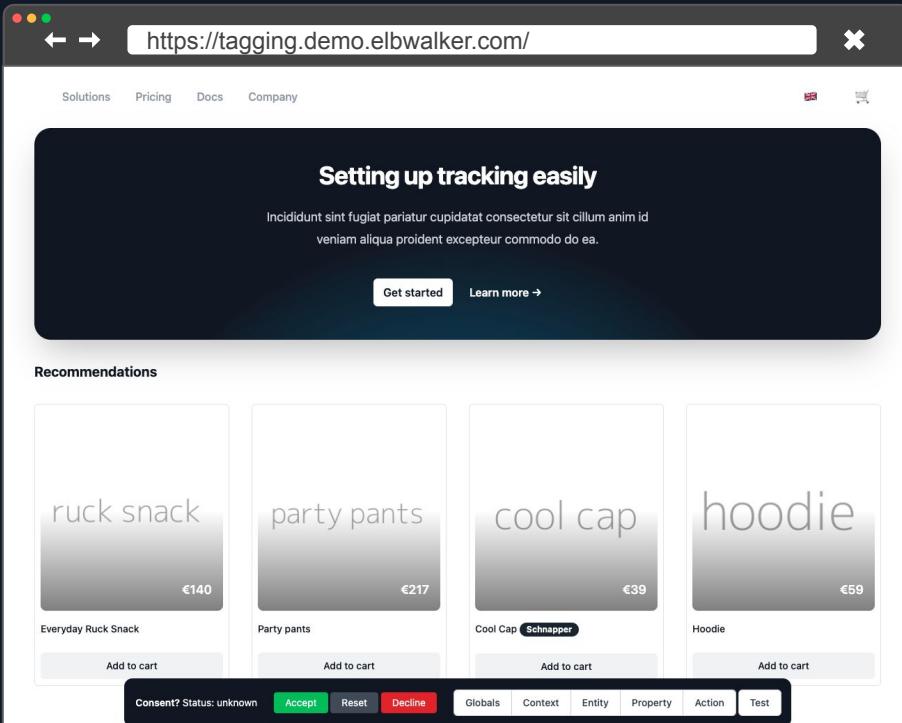


```
{  
  globals: {  
    language: "en"  
  },  
  context: {  
    position: ["hero", 0]  
  },  
  event: "promotion view",  
  
  data: {  
    name: "How to tag"  
  }  
}
```



Data collection with walkerOS

Just try it



Tagging playground

~60 data-elb attributes

Enhanced ecommerce

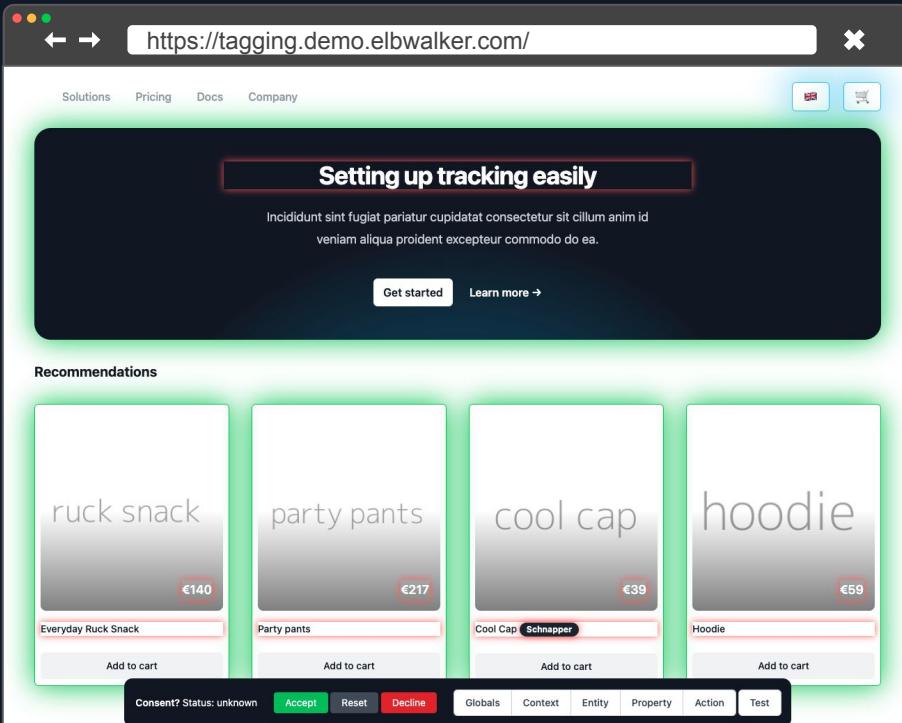
Consent management

User identification

5 Destinations

Data collection with walkerOS

Just try it



Tagging playground

~60 data-elb attributes

Enhanced ecommerce

Consent management

User identification

5 Destinations

Only HTML Attributes

How to tag elblish

`data-elb="entity"`

Sets the event scope
product, order

`data-elbaction="trigger:action"`

Initializes a trigger to fire an action
click:add, load:complete

`data-elb-entity="key:value"`

Defines the data of a specific entity
name:Shirt, revenue:9001

Only HTML Attributes

How to tag elblish

data-elb="entity"

Sets the event scope
product, order

data-elbcontext="test:engagement"

Added only to related events
list:Inspo, shopping:complete

data-elbaction="trigger:action"

Initializes a trigger to fire an action
click:add, load:complete

data-elbglobals="key:value"

Added to every single event
language:en, pagetype:demo

data-elb-entity="key:value"

Defines the data of a specific entity
name:Shirt, revenue:9001

data-elblink="parent:id"

For more advanced edge cases
Ask me later...

Only HTML Attributes

How to tag elkish

`data-elb="promotion"`

Setting up tracking easily

Incididunt sint fugiat pariatur cupidatat consectetur sit cillum anim

Only HTML Attributes

How to tag elkish

`data-elbaction="visible:view"`

Setting up tracking easily

Incididunt sint fugiat pariatur cupidatat consectetur sit cillum anim
id veniam aliqua proident excepteur commodo do ea.

[Get started](#)

[Learn more →](#)

Only HTML Attributes

How to tag elkish

`data-elbaction="visible:view"`

Setting up tracking easily

Incididunt sint fugiat pariatur cupidatat consectetur sit cillum anim
id veniam aliqua proident excepteur commodo do ea.

[Get started](#)

[Learn more →](#)

`event: promotion view`

Only HTML Attributes

How to tag elkish

`data-elb-promotion="name: ..."`

Setting up tracking easily

Incididunt sint fugiat pariatur cupidatat consectetur sit cillum anim
id veniam aliqua proident excepteur commodo do ea.

[Get started](#)

[Learn more →](#)

`event: promotion view
data`

`name: Setting up tracking easily`

Only HTML Attributes

How to tag elkish

`data-elbcontext="position:left"`

Setting up tracking easily

Incididunt sint fugiat pariatur cupidatat consectetur sit cillum anim
id veniam aliqua proident excepteur commodo do ea.

[Get started](#)

[Learn more →](#)

```
event: promotion view
data
name: Setting up tracking easily
context
position: left
```

Only HTML Attributes

How to tag elkish

`data-elbglobals="language:en"`

Setting up tracking easily

Incididunt sint fugiat pariatur cupidatat consectetur sit cillum anim
id veniam aliqua proident excepteur commodo do ea.

[Get started](#)

[Learn more →](#)

```
event: promotion view
data
  name: Setting up tracking easily
  context
  position: left

globals
  language: en
```

Only HTML Attributes

How to tag elkish

`data-elbcontext="font:beautiful"`

Setting up tracking easily

*Incididunt sint fugiat pariatur cupidatat consectetur sit cillum
anim id veniam aliqua proident excepteur commodo do ea.*

[Get started](#)

[Learn more →](#)

```
event: promotion view
data
name: Setting up tracking easily
context
position: left
font: beautiful
globals
language: en
```

Only HTML Attributes

How to tag elkish

`data-elbaction="click:start"`

Setting up tracking easily

*Incididunt sint fugiat pariatur cupidatat consectetur sit cillum
anim id veniam aliqua proident excepteur commodo do ea.*

[Get started](#)

[Learn more →](#)

```
event: promotion start
data
  name: Setting up tracking easily
  context
  position: left
  font: beautiful
  globals
  language: en
```

The event model

Law and order

```
...
  walker event
{
  event: 'promotion view', // Combination of entity and action
  data: {
    // Arbitrary set properties with the data-elb-promotion attribute
    name: 'Setting up tracking easily',
    category: 'analytics',
  },
  context: {
    // Related properties defined with the data-elbcontext attribute
    test: ['engagement', 0] // Value and order
  },
  custom: {}, // Additional custom data for individual setups
  globals: {
    // General Properties defined with the data-elbglobals attribute
    language: 'en'
  },
  user: {
    // Stored user ids (manually added once)
    id: 'userid',
    device: 'cookieid',
    session: 'sessionid',
  },
  nested: [], // All nested entities within the promotion
  consent: { functional: true }, // Status of the consent state(s)
  id: '1647261462000-01b5e2-5', // Timestamp, group & count of the event
  trigger: 'visible', // Name of the trigger that fired
  entity: 'promotion', // Entity name
  action: 'view', // Entity action
  timestamp: 1647261462000, // Time when the event fired
  timing: 3.14, // How long it took from the page load to trigger the event
  group: '01b5e2', // Random group id for all events on a page
  count: 2, // Incremental counter of the events on a page
  version: {
    // Helpful when working with raw data
    client: '1.0.0', // Semantic version of the used client
    tagging: 42, // A version number of the then used tagging status
  },
  source: {
    // Origins of the event
    type: 'web', // Source type of the event
    id: 'https://github.com/elbwalker/walker0S', // Source id of the event's origin (url)
    previous_id: 'https://www.elbwalker.com/' // Previous source id (referrer)
  },
  walker: true, // Flag to filter events
}
```

```
event: promotion start
data
  name: Setting up tracking easily
context
  position: left
  font: beautiful
globals
  language: en
```

The event model

Law and order

```
...
  walker event
{
  event: 'promotion view', // Combination of entity and action
  data: {
    // Arbitrary set properties with the data-elb-promotion attribute
    name: 'Setting up tracking easily',
    category: 'analytics',
  },
  context: {
    // Related properties defined with the data-elbcontext attribute
    test: ['engagement', 0] // Value and order
  },
  custom: {}, // Additional custom data for individual setups
  globals: {
    // General Properties defined with the data-elbglobals attribute
    language: 'en'
  },
  user: {
    // Stored user ids (manually added once)
    id: 'userid',
    device: 'cookieid',
    session: 'sessionid',
  },
  nested: [], // All nested entities within the promotion
  consent: { functional: true }, // Status of the consent state(s)
  id: '1647261462000-01b5e2-5', // Timestamp, group & count of the event
  trigger: 'visible', // Name of the trigger that fired
  entity: 'promotion', // Entity name
  action: 'view', // Entity action
  timestamp: 1647261462000, // Time when the event fired
  timing: 3.14, // How long it took from the page load to trigger the event
  group: '01b5e2', // Random group id for all events on a page
  count: 2, // Incremental counter of the events on a page
  version: {
    // Helpful when working with raw data
    client: '1.0.0', // Semantic version of the used client
    tagging: 42, // A version number of the then used tagging status
  },
  source: {
    // Origins of the event
    type: 'web', // Source type of the event
    id: 'https://github.com/elbwalker/walkerOS', // Source id of the event's origin (url)
    previous_id: 'https://www.elbwalker.com/' // Previous source id (referrer)
  },
  walker: true, // Flag to filter events
}
```

timestamp: 1647261462000
timing: 3.14
group: 01b5e2
count: 2
version
 client: 1.0.0
 tagging: 42

 DevOps &
Developer

 Data
Analyst

 Product
Owner

 Growth
Manager

 Data
Scientist

The event model

Law and order

```
...
  walker event
{
  event: 'promotion view', // Combination of entity and action
  data: {
    // Arbitrary set properties with the data-elb-promotion attribute
    name: 'Setting up tracking easily',
    category: 'analytics',
  },
  context: {
    // Related properties defined with the data-elbcontext attribute
    test: ['engagement', 0] // Value and order
  },
  custom: {}, // Additional custom data for individual setups
  globals: {
    // General Properties defined with the data-elbglobals attribute
    language: 'en'
  },
  user: {
    // Stored user ids (manually added once)
    id: 'userid',
    device: 'cookieid',
    session: 'sessionid',
  },
  nested: [], // All nested entities within the promotion
  consent: { functional: true }, // Status of the consent state(s)
  id: '1647261462000-01b5e2-5', // Timestamp, group & count of the event
  trigger: 'visible', // Name of the trigger that fired
  entity: 'promotion', // Entity name
  action: 'view', // Entity action
  timestamp: 1647261462000, // Time when the event fired
  timing: 3.14, // How long it took from the page load to trigger the event
  group: '01b5e2', // Random group id for all events on a page
  count: 2, // Incremental counter of the events on a page
  version: {
    // Helpful when working with raw data
    client: '1.0.0', // Semantic version of the used client
    tagging: 42, // A version number of the then used tagging status
  },
  source: {
    // Origins of the event
    type: 'web', // Source type of the event
    id: 'https://github.com/elbwalker/walkerOS', // Source id of the event's origin (url)
    previous_id: 'https://www.elbwalker.com/' // Previous source id (referrer)
  },
  walker: true, // Flag to filter events
}
```

user

id: userid
device: cookieid
session: sessionid

The event model

Law and order

```
...
  walker event
{
  event: 'promotion view', // Combination of entity and action
  data: {
    // Arbitrary set properties with the data-elb-promotion attribute
    name: 'Setting up tracking easily',
    category: 'analytics',
  },
  context: {
    // Related properties defined with the data-elbcontext attribute
    test: ['engagement', 0] // Value and order
  },
  custom: {}, // Additional custom data for individual setups
  globals: {
    // General Properties defined with the data-elbglobals attribute
    language: 'en'
  },
  user: {
    // Stored user ids (manually added once)
    id: 'userid',
    device: 'cookieid',
    session: 'sessionid',
  },
  nested: [], // All nested entities within the promotion
  consent: { functional: true }, // Status of the consent state(s)
  id: '1647261462000-01b5e2-5', // Timestamp, group & count of the event
  trigger: 'visible', // Name of the trigger that fired
  entity: 'promotion', // Entity name
  action: 'view', // Entity action
  timestamp: 1647261462000, // Time when the event fired
  timing: 3.14, // How long it took from the page load to trigger the event
  group: '01b5e2', // Random group id for all events on a page
  count: 2, // Incremental counter of the events on a page
  version: {
    // Helpful when working with raw data
    client: '1.0.0', // Semantic version of the used client
    tagging: 42, // A version number of the then used tagging status
  },
  source: {
    // Origins of the event
    type: 'web', // Source type of the event
    id: 'https://github.com/elbwalker/walkerOS', // Source id of the event's origin (url)
    previous_id: 'https://www.elbwalker.com/' // Previous source id (referrer)
  },
  walker: true, // Flag to filter events
}
```

consent

functional: true
marketing: false

The event model

Law and order

```
...
  walker event
{
  event: 'promotion view', // Combination of entity and action
  data: {
    // Arbitrary set properties with the data-elb-promotion attribute
    name: 'Setting up tracking easily',
    category: 'analytics',
  },
  context: {
    // Related properties defined with the data-elbcontext attribute
    test: ['engagement', 0] // Value and order
  },
  custom: {}, // Additional custom data for individual setups
  globals: {
    // General Properties defined with the data-elbglobals attribute
    language: 'en'
  },
  user: {
    // Stored user ids (manually added once)
    id: 'userid',
    device: 'cookieid',
    session: 'sessionid',
  },
  nested: [], // All nested entities within the promotion
  consent: { functional: true }, // Status of the consent state(s)
  id: '1647261462000-01b5e2-5', // Timestamp, group & count of the event
  trigger: 'visible', // Name of the trigger that fired
  entity: 'promotion', // Entity name
  action: 'view', // Entity action
  timestamp: 1647261462000, // Time when the event fired
  timing: 3.14, // How long it took from the page load to trigger the event
  group: '01b5e2', // Random group id for all events on a page
  count: 2, // Incremental counter of the events on a page
  version: {
    // Helpful when working with raw data
    client: '1.0.0', // Semantic version of the used client
    tagging: 42, // A version number of the then used tagging status
  },
  source: {
    // Origins of the event
    type: 'web', // Source type of the event
    id: 'https://github.com/elbwalker/walkerJS', // Source id of the event's origin (url)
    previous_id: 'https://www.elbwalker.com/' // Previous source id (referrer)
  },
  walker: true, // Flag to filter events
}
```

```
event: page view,
data
domain 127.0.0.1
id: /demo
title: Tagging playground
source
type: web
```

The event model

Law and order

```
...
  walker event
{
  event: 'promotion view', // Combination of entity and action
  data: {
    // Arbitrary set properties with the data-elb-promotion attribute
    name: 'Setting up tracking easily',
    category: 'analytics',
  },
  context: {
    // Related properties defined with the data-elbcontext attribute
    test: ['engagement', 0] // Value and order
  },
  custom: {}, // Additional custom data for individual setups
  globals: {
    // General Properties defined with the data-elbglobals attribute
    language: 'en'
  },
  user: {
    // Stored user ids (manually added once)
    id: 'userid',
    device: 'cookieid',
    session: 'sessionid',
  },
  nested: [], // All nested entities within the promotion
  consent: { functional: true }, // Status of the consent state(s)
  id: '1647261462000-01b5e2-5', // Timestamp, group & count of the event
  trigger: 'visible', // Name of the trigger that fired
  entity: 'promotion', // Entity name
  action: 'view', // Entity action
  timestamp: 1647261462000, // Time when the event fired
  timing: 3.14, // How long it took from the page load to trigger the event
  group: '01b5e2', // Random group id for all events on a page
  count: 2, // Incremental counter of the events on a page
  version: {
    // Helpful when working with raw data
    client: '1.0.0', // Semantic version of the used client
    tagging: 42, // A version number of the then used tagging status
  },
  source: {
    // Origins of the event
    type: 'web', // Source type of the event
    id: 'https://github.com/elbwalker/walkerOS', // Source id of the event's origin (url)
    previous_id: 'https://www.elbwalker.com/' // Previous source id (referrer)
  },
  walker: true, // Flag to filter events
}
```

event: session start
data
id: s3ss10n
marketing: true,
campaign talk
source
type: web

The event model

Law and order

```
...
  walker event
{
  event: 'promotion view', // Combination of entity and action
  data: {
    // Arbitrary set properties with the data-elb-promotion attribute
    name: 'Setting up tracking easily',
    category: 'analytics',
  },
  context: {
    // Related properties defined with the data-elbcontext attribute
    test: ['engagement', 0] // Value and order
  },
  custom: {}, // Additional custom data for individual setups
  globals: {
    // General Properties defined with the data-elbglobals attribute
    language: 'en'
  },
  user: {
    // Stored user ids (manually added once)
    id: 'userid',
    device: 'cookieid',
    session: 'sessionid',
  },
  nested: [], // All nested entities within the promotion
  consent: { functional: true }, // Status of the consent state(s)
  id: '1647261462000-01b5e2-5', // Timestamp, group & count of the event
  trigger: 'visible', // Name of the trigger that fired
  entity: 'promotion', // Entity name
  action: 'view', // Entity action
  timestamp: 1647261462000, // Time when the event fired
  timing: 3.14, // How long it took from the page load to trigger the event
  group: '01b5e2', // Random group id for all events on a page
  count: 2, // Incremental counter of the events on a page
  version: {
    // Helpful when working with raw data
    client: '1.0.0', // Semantic version of the used client
    tagging: 42, // A version number of the then used tagging status
  },
  source: {
    // Origins of the event
    type: 'web', // Source type of the event
    id: 'https://github.com/elbwalker/walkerJS', // Source id of the event's origin (url)
    previous_id: 'https://www.elbwalker.com/' // Previous source id (referrer)
  },
  walker: true, // Flag to filter events
}
```

event: session end
data
id: s3ss10n
duration: 2520
stage: checkout
source
type: server

The event model

Law and order

```
...
  walker event
{
  event: 'promotion view', // Combination of entity and action
  data: {
    // Arbitrary set properties with the data-elb-promotion attribute
    name: 'Setting up tracking easily',
    category: 'analytics',
  },
  context: {
    // Related properties defined with the data-elbcontext attribute
    test: ['engagement', 0] // Value and order
  },
  custom: {}, // Additional custom data for individual setups
  globals: {
    // General Properties defined with the data-elbglobals attribute
    language: 'en'
  },
  user: {
    // Stored user ids (manually added once)
    id: 'userid',
    device: 'cookieid',
    session: 'sessionid',
  },
  nested: [], // All nested entities within the promotion
  consent: { functional: true }, // Status of the consent state(s)
  id: '1647261462000-01b5e2-5', // Timestamp, group & count of the event
  trigger: 'visible', // Name of the trigger that fired
  entity: 'promotion', // Entity name
  action: 'view', // Entity action
  timestamp: 1647261462000, // Time when the event fired
  timing: 3.14, // How long it took from the page load to trigger the event
  group: '01b5e2', // Random group id for all events on a page
  count: 2, // Incremental counter of the events on a page
  version: {
    // Helpful when working with raw data
    client: '1.0.0', // Semantic version of the used client
    tagging: 42, // A version number of the then used tagging status
  },
  source: {
    // Origins of the event
    type: 'web', // Source type of the event
    id: 'https://github.com/elbwalker/walkerJS', // Source id of the event's origin (url)
    previous_id: 'https://www.elbwalker.com/' // Previous source id (referrer)
  },
  walker: true, // Flag to filter events
}
```

```
event: screen view
data
  id: abc
  type: full
  name: welcome
source
  type: app
```

The event model

Law and order

```
...
  walker event
{
  event: 'promotion view', // Combination of entity and action
  data: {
    // Arbitrary set properties with the data-elb-promotion attribute
    name: 'Setting up tracking easily',
    category: 'analytics',
  },
  context: {
    // Related properties defined with the data-elbcontext attribute
    test: ['engagement', 0] // Value and order
  },
  custom: {}, // Additional custom data for individual setups
  globals: {
    // General Properties defined with the data-elbglobals attribute
    language: 'en'
  },
  user: {
    // Stored user ids (manually added once)
    id: 'userid',
    device: 'cookieid',
    session: 'sessionid',
  },
  nested: [], // All nested entities within the promotion
  consent: { functional: true }, // Status of the consent state(s)
  id: '1647261462000-01b5e2-5', // Timestamp, group & count of the event
  trigger: 'visible', // Name of the trigger that fired
  entity: 'promotion', // Entity name
  action: 'view', // Entity action
  timestamp: 1647261462000, // Time when the event fired
  timing: 3.14, // How long it took from the page load to trigger the event
  group: '01b5e2', // Random group id for all events on a page
  count: 2, // Incremental counter of the events on a page
  version: {
    // Helpful when working with raw data
    client: '1.0.0', // Semantic version of the used client
    tagging: 42, // A version number of the then used tagging status
  },
  source: {
    // Origins of the event
    type: 'web', // Source type of the event
    id: 'https://github.com/elbwalker/walkerOS', // Source id of the event's origin (url)
    previous_id: 'https://www.elbwalker.com/' // Previous source id (referrer)
  },
  walker: true, // Flag to filter events
}
```

- ✓ First party
- ✓ Source of truth
- ✓ Structured flexibility
- ✓ Different sources
- ✓ Vendor agnostic
- ✓ Industry independent

The event model

Law and order

```
...
  walker event
{
  event: 'promotion view', // Combination of entity and action
  data: {
    // Arbitrary set properties with the data-elb-promotion attribute
    name: 'Setting up tracking easily',
    category: 'analytics',
  },
  context: {
    // Related properties defined with the data-elbcontext attribute
    test: ['engagement', 0] // Value and order
  },
  custom: {}, // Additional custom data for individual setups
  globals: {
    // General Properties defined with the data-elbglobals attribute
    language: 'en'
  },
  user: {
    // Stored user ids (manually added once)
    id: 'userid',
    device: 'cookieid',
    session: 'sessionid',
  },
  nested: [], // All nested entities within the promotion
  consent: { functional: true }, // Status of the consent state(s)
  id: '1647261462000-01b5e2-5', // Timestamp, group & count of the event
  trigger: 'visible', // Name of the trigger that fired
  entity: 'promotion', // Entity name
  action: 'view', // Entity action
  timestamp: 1647261462000, // Time when the event fired
  timing: 3.14, // How long it took from the page load to trigger the event
  group: '01b5e2', // Random group id for all events on a page
  count: 2, // Incremental counter of the events on a page
  version: {
    // Helpful when working with raw data
    client: '1.0.0', // Semantic version of the used client
    tagging: 42, // A version number of the then used tagging status
  },
  source: {
    // Origins of the event
    type: 'web', // Source type of the event
    id: 'https://github.com/elbwalker/walkerOS', // Source id of the event's origin (url)
    previous_id: 'https://www.elbwalker.com/' // Previous source id (referrer)
  },
  walker: true, // Flag to filter events
}
```

- ✓ First party
- ✓ Source of truth
- ✓ Structured flexibility
- ✓ Different sources
- ✓ Vendor agnostic
- ✓ Industry independent

**Nothing
has been tracked yet!**

Data **infrastructure**

How to get from A to Z

Clients

Web

Data **infrastructure**

How to get from A to Z

Clients

Destinations



Data infrastructure

How to get from A to Z

Clients

Web

Destinations

Marketing



Analytics



Data infrastructure

How to get from A to Z

Clients

Stacks

Destinations

Web



Marketing



Analytics



Data infrastructure

How to get from A to Z

Clients

Stacks

Destinations

Web



Marketing



Analytics



Data infrastructure

How to get from A to Z

Clients

Web

App

Server

Stacks



Destinations

Marketing



Analytics



Data infrastructure

How to get from A to Z

Clients

Web

App

Server

Stacks



Destinations

Marketing



Analytics



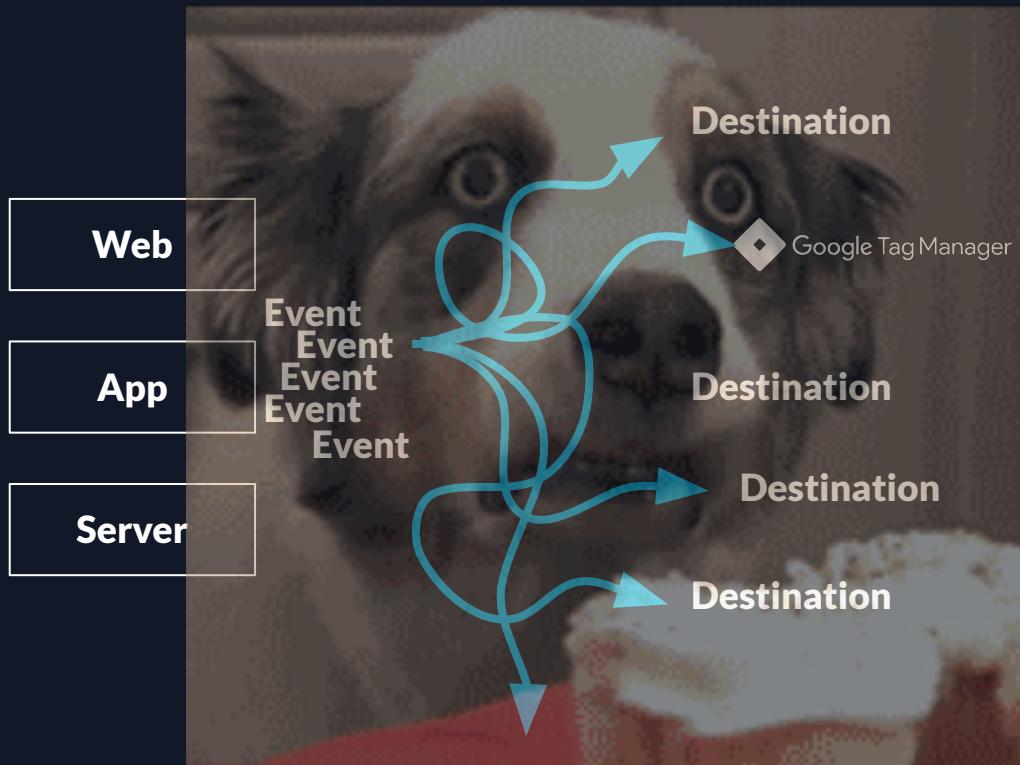
Database



Data infrastructure

How to get from A to Z

Clients



Stacks

Destinations

Marketing

▲ Google Ads ∞ Meta

Analytics

● Google Analytics 4 SNOWPLOW
PIWIK PRO

Database

● Google BigQuery snowflake

Data infrastructure

How to get from A to Z

Clients

Stacks

Destinations

Web



App

Server

Marketing



Analytics



Database



Data infrastructure

How to get from A to Z

Clients

Stacks

Destinations

Web

App

Server



Server-side
Google Tag Manager

Marketing



Analytics



Database



Data collection with walkerOS

A few pipeline examples



tagging.html

```
<body data-elbglobals="language:en">
  <div data-elbcontext="test:engagement">
    <div data-elb="promotion" data-elbaction="visible:view">
      <h1 data-elb-promotion="name:Setting up tracking easily">
        Setting up tracking easily
      </h1>
      <p data-elb-promotion="category:analytics">Analytics</p>
    </div>
  </div>
</body>
```

Data collection with walkerOS

A few pipeline examples

• • •

tagging.html

```
<body data-elbglobals="language:en">
  <div data-elbcontext="test:engagement">
    <div data-elb="promotion" data-elbaction="visible:view">
      <h1 data-elb-promotion="name:Setting up tracking easily">
        Setting up tracking easily
      </h1>
      <p data-elb-promotion="category:analytics">Analytics</p>
    </div>
  </div>
</body>
```

Data collection with walkerOS

A few pipeline examples

The screenshot shows the 'walker.js' web client interface. At the top, there's a header with a close button ('X'), the file name 'walker.js', a folder icon, a 'Save' button, and a three-dot menu icon. Below the header, the title 'Tag Configuration' is displayed. Under 'Tag Type', a 'Custom HTML' tag is selected, indicated by a highlighted box and the text 'Custom HTML Tag'. In the 'HTML' section, there's a question mark icon and a code editor containing the following script:

```
1 <script>
2 var elb = function(){ window.elbLayer.push(arguments)};
3 var elbwalker = {{walker.js}}({ default: true });
4 </script>
```

Tagged
Website

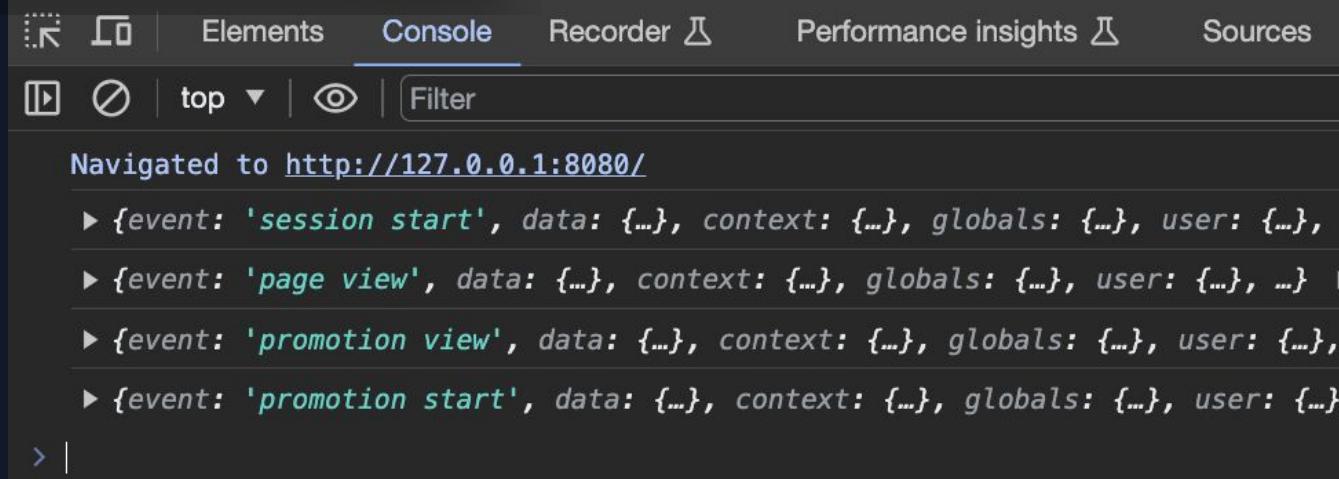
walker.js
web client

Data collection with walkerOS

A few pipeline examples

console destination

```
elb('walker destination', { push: console.log });
```



The screenshot shows the browser's developer tools open to the 'Console' tab. The tab bar includes 'Elements', 'Console' (which is underlined), 'Recorder', 'Performance insights', and 'Sources'. Below the tab bar are buttons for 'top' and 'Filter'. The main area displays the following log output:

```
Navigated to http://127.0.0.1:8080/
▶ {event: 'session start', data: {...}, context: {...}, globals: {...}, user: {...},
▶ {event: 'page view', data: {...}, context: {...}, globals: {...}, user: {...}, ...
▶ {event: 'promotion view', data: {...}, context: {...}, globals: {...}, user: {...},
▶ {event: 'promotion start', data: {...}, context: {...}, globals: {...}, user: {...}
> |
```



Data collection with walkerOS

A few pipeline examples



Connected
127.0.0.1

1 Google tag found GTM-WCFR4W5

Summary Event: promotion view

walkerOS tagging demo

9 promotion start

8 promotion view

7 Window Loaded

6 DOM Ready

5 page view

4 session start

3 Container Loaded

2 Initialization

1 Consent Initialization

API Call

```
dataLayer.push({
  event: "promotion view",
  data: {category: "analytics", name: "Setting up tracking easily"},
  context: {test: ["engagement", 0], category: ["analytics", 0]},
  globals: {language: "en", cart_value: 249},
  user: {},
  nested: [],
  consent: {},
  id: "1699616357157-sgyp0z-3",
  trigger: "visible",
  entity: "promotion",
  action: "view",
  timestamp: 1699616357157,
  timing: 1.69,
  group: "sgyp0z",
  count: 3,
  version: {config: 1, walker: 1.6},
  source: {
    type: 1,
    id: "http://127.0.0.1:8080/?gtm_debug=1699616355459",
    previous_id: "https://tagassistant.google.com/"
  },
  walker: true,
  gtm.uniqueEventId: 14
})
```

Data collection with walkerOS

A few pipeline examples

The screenshot shows a web-based configuration interface for a 'walker.js destination Piwik PRO'. The title bar says 'walker.js destination Piwik PRO'. On the right, there's a 'Save' button and a three-dot menu icon. The main area is titled 'Tag Configuration' under 'Tag Type' and shows 'Custom HTML' selected. Below that is a code editor with syntax highlighting for JavaScript. The code defines a function 'elb' that takes a 'walker destination' object and a 'Piwik PRO' object as arguments. It includes settings for consent states, a custom appId ('xxxxxxxx-xxxx-xxxx-xxxx-xxxxxxxxxxxx'), a URL ('https://elbwalker.piwik.pro/'), and a mapping object that maps an event ('complete') to a goal ('data.total').

```
1 <script>
2 elb("walker destination", {{walker.js destination Piwik PRO}}, {
3   consent: { functional: true }, // Necessary consent states
4   custom: {
5     appId: 'xxxxxxxx-xxxx-xxxx-xxxx-xxxxxxxxxxxx', // Id of the site
6     url: 'https://elbwalker.piwik.pro/'}, // Same address as the login
7   },
8   loadScript: true, // Load additional required scripts on init
9   mapping: {
10     'order': {
11       'complete': {
12         custom: {
13           goalId: 'xxxx-xxxx-xxxx-xxxx-xxxx', // Count the event as a goal
14           goalValue: 'data.total', // Property to be used as goal value
15         },
16       },
17     },
18   },
19 })
```



Data collection with **walkerOS**

A few pipeline examples



Firebase stack

```
import { firebaseStack } from '@elbwalker/stack-firebase';

const { elb, push } = firebaseStack({});

export const ingest = push({ cors: true, region: 'europe-west1' });
```

Tagged
Website

walker.js
web client

Firebase
Stack

Data collection with walkerOS

A few pipeline examples



Firebase stack

```
import { firebaseStack } from '@elbwalker/stack-firebase';
import { destinationBigQuery }
  from '@elbwalker/destination-node-bigquery';

const { elb, push } = firebaseStack({});

elb('walker destination', destinationBigQuery, {
  custom: { projectId: 'my-own' },
});

export const ingest = push({ cors: true, region: 'europe-west1' });
```



Data collection with walkerOS

A few pipeline examples



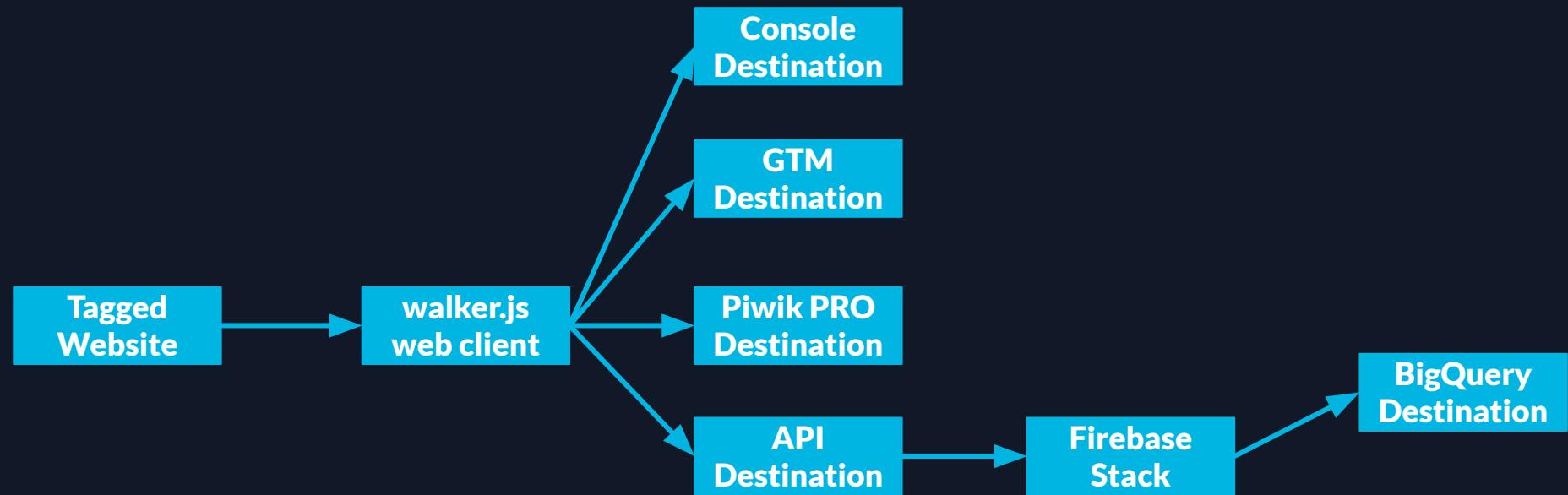
API destination

```
elb('walker destination', api, {
  consent: { functional: true },
  custom: {
    url: 'http://127.0.0.1:5001.firebaseio/europe-west1/ingest',
  },
});
```



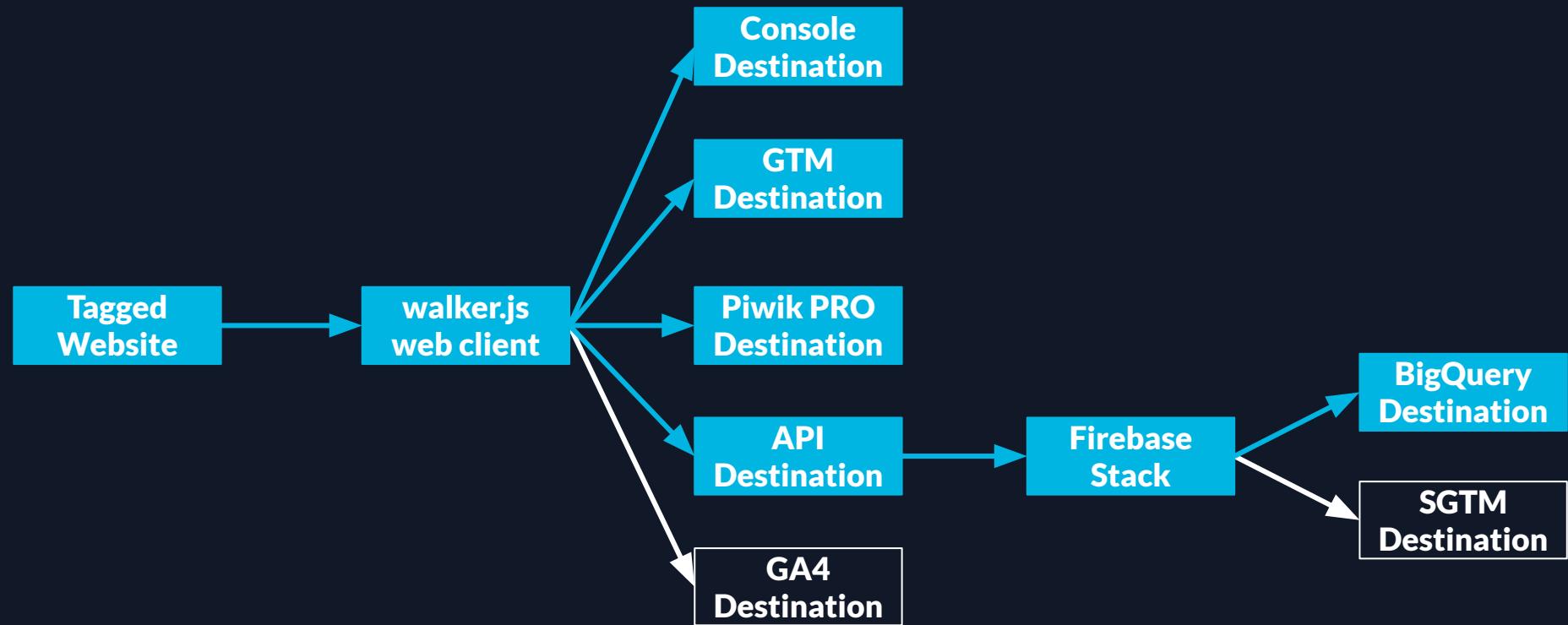
Data collection with walkerOS

A complete pipeline example



Data collection with walkerOS

A complete pipeline example



Data collection with walkerOS

Mapping helps to translate

```
event: promotion view
data:
  name: Setting up tracking
  category: analytics
globals:
  language: en
context:
  test: engagement
```

Data collection with **walkerOS**

Mapping helps to translate

```
event: promotion view
```

```
data
```

```
  name: Setting up tracking
```

```
  category: analytics
```

```
globals
```

```
  language: en
```

```
context
```

```
  test: engagement
```

```
promotion_view
```

```
trackContentImpression
```

```
view_promotion
```

```
trackPromotionView
```



Data collection with walkerOS

Contracts ensure quality

```
event: promotion view
data
  name: Setting up tracking
  category: analytics
globals
  language: en
context
  test: engagement
```

```
contract-analyst.ts

● ● ●

const analyst: Schema.Contract = {
  promotion: { 'view': {
    data: { schema: { name: { required: true } } }
  } }
};
```

```
contract-scientist.ts

● ● ●

const scientist: Schema.Contract = {
  '*': { '*': {
    globals: { schema: { cart_value: { type: 'number' } } }
  } }
};
```

Show me!

No **big** deal

It's the little things in life

~11Kb

Runs fast

Resource-saving

How to get **started**

Next steps

**What stops you
from using it?**



Thank you!



GA4 Destination

```
elb("walker destination", {
  consent: { marketing: true, ga4: true },
  custom: {
    include: ["globals"],
    measurementId: "G-M34SUR3M3",
    params: { user_id: "user.id" },
  },
  loadScript: true,
  mapping: {
    product: {
      add: {
        name: "add_to_cart",
        custom: {
          include: ["all"],
          items: {
            params: {
              item_id: "data.id",
              item_category: "context.category.0",
              quantity: { default: 1, key: "data.quantity" }
            }
          },
          params: { value: "data.price" },
        }
      }
    },
    order: { complete: { name: "purchase", custom: { ... } } }
  }
});
```

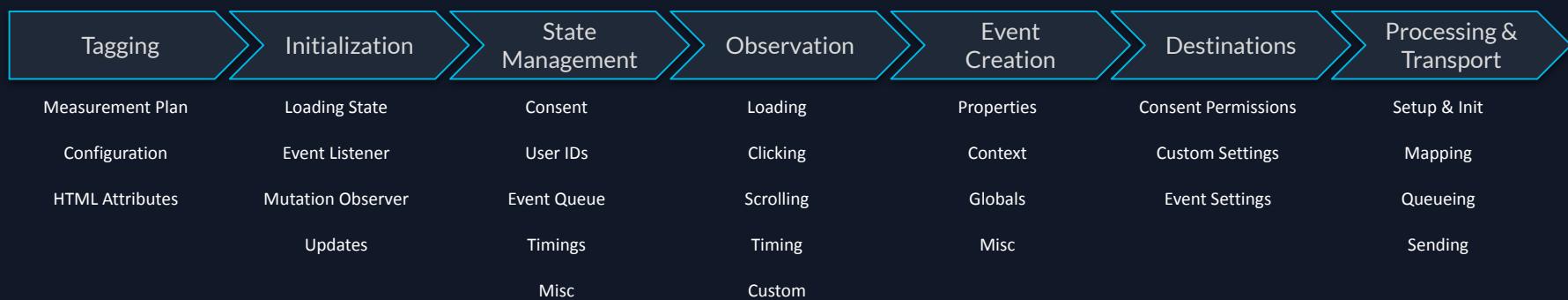
How to get **started**

Next steps

I'm bad with
documentation

Data collection with `walker.js`

Once upon a `dataLayer.push`



Data collection with walkerOS

Show me



tagging.html

```
<body data-elbglobals="language:en">
  <div data-elbcontext="test:engagement">
    <div data-elb="promotion" data-elbaction="visible:view">
      <h1 data-elb-promotion="name:Setting up tracking easily">
        Setting up tracking easily
      </h1>
      <p data-elb-promotion="category:analytics">Analytics</p>
    </div>
  </div>
</body>
```



Console Destination

```
elb('walker destination', { push: console.log });
```



API Destination

```
elb('walker destination', api, {
  custom: { url: 'http://localhost/europe-west1/ingest' },
});
```



GTM Destination

```
import destinationGTM from
  `@elbwalker/destination-web-google-gtm`;

elb('walker destination', destinationGTM, {
  consent: { functional: true },
  loadScript: true,
  custom: {
    containerId: 'GTM-WCFR4W5',
  },
});
```



SGTM Destination

```
elb('walker destination', api, {
  consent: { functional: true },
  custom: {
    url: 'https://server-side-tagging-ybbfvkxhia-ey.a.run.app/quokka',
    transport: 'beacon'
  }
});
```

Data collection with walkerOS

GA4 Destination

```
elb("walker destination", {  
  consent: { marketing: true, ga4: true },  
  custom: {  
    include: ["globals"],  
    measurementId: "G-M34SUR3M3",  
    params: { user_id: "user.id" },  
  },  
  loadScript: true,  
  mapping: {  
    product: {  
      add: {  
        name: "add_to_cart",  
        custom: {  
          include: ["all"],  
          items: {  
            params: {  
              item_id: "data.id",  
              item_category: "context.category.0",  
              quantity: { default: 1, key: "data.quantity" }  
            },  
            params: { value: "data.price" },  
          }  
        }  
      },  
      order: { complete: { name: "purchase", custom: { ... } } }  
    }  
  },  
});
```

Show me

Consent Command

```
elb('walker consent', { functional: true, marketing: true });
```

User Command

```
elb('walker user', {  
  id: storageRead('elb_user_id'),  
  session: storageRead('elb_user_session'),  
  device: storageRead('elb_user_device'),  
});
```

Utils

```
const event validateEvent(request.body, contracts);  
const session = sessionStart(options);  
const visible = isVisible(element);
```

Data collection with walkerOS

Show me

• • • Data Contract

```
const analystContract: Schema.Contract = {  
  product: { '*': {  
    data: { schema: { name: { required: true } } } }  
  } }  
};
```

• • • Data Contract

```
const scientistContract: Schema.Contract = {  
  '*': { '*': {  
    globals: { schema: { cart_value: { type: 'number' } } } } }  
};
```

• • • Firebase Stack

```
elb('walker destination', destinationBigQuery, {  
  custom: { projectId: 'eventpipe-f9979' },  
});
```

• • • Firebase Stack

```
import { firebaseStack } from '@elbwalker/stack-firebase';  
  
const { elb, push } = firebaseStack({  
  client: {  
    contracts: [analystContract, scientistContract],  
  },  
});
```

• • • Firebase Stack

```
export const ingest = push({ cors: true, region: 'europe-west1' });
```