



# Hi, I'm **Yogev Raved**

Building end-to-end user experiences for complex systems

# What makes me tick

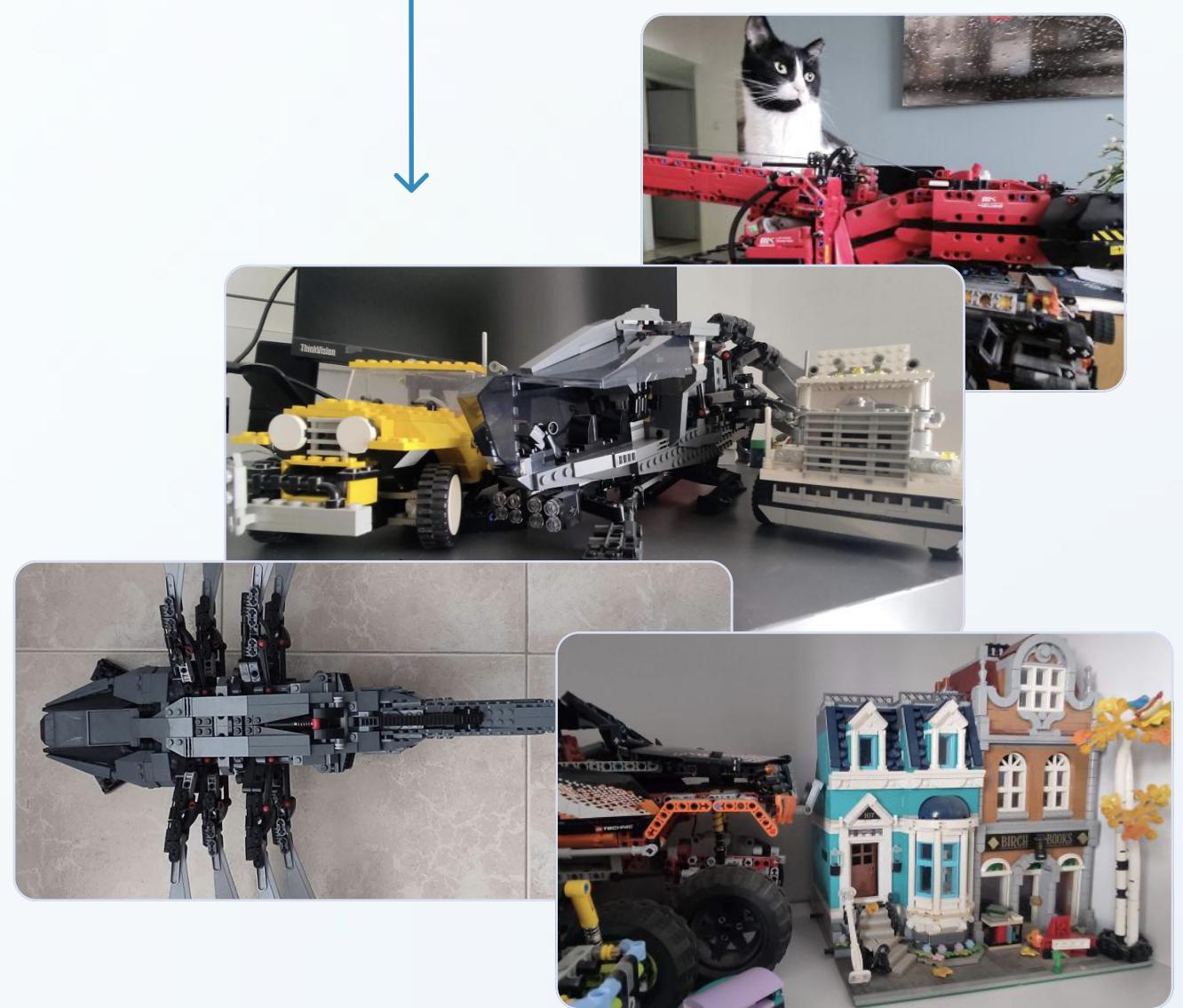
Dragging my family  
to hiking trips



Thousands of wasted hours  
on city building games



Geeking out  
with Lego



# Every design has a story



# Config-templates design revamp

Data Grid View			
Row ID	Name	Author	Last Editor
1	Config-template-name	User Name	User Name 2
2	Config-template-name	User Name	User Name 2
3	Config-template-name	User Name	User Name 2
4	Config-template-name	User Name	User Name 2
5	Config-template-name	User Name	User Name 2
6	Config-template-name	User Name	User Name 2
7	Config-template-name	User Name	User Name 2
8	Config-template-name	User Name	User Name 2
9	Config-template-name	User Name	User Name 2
10	Config-template-name	User Name	User Name 2
11	Config-template-name	User Name	User Name 2
12	Config-template-name	User Name	User Name 2
13	Config-template-name	User Name	User Name 2
14	Config-template-name	User Name	User Name 2
15	Config-template-name	User Name	User Name 2
16	Config-template-name	User Name	User Name 2
17	Config-template-name	User Name	User Name 2
18	Config-template-name	User Name	User Name 2
19	Config-template-name	User Name	User Name 2
20	Config-template-name	User Name	User Name 2



# Marketing campaigns management tool

The image displays five distinct sections from a digital marketing platform, each featuring a different campaign or promotional banner:

- Top Left:** A screenshot of a landing page for "Award Winning Online Protection". It includes a "PC Doctor" software box, a "FREE" offer button, and a "START YOUR TEST" button.
- Top Middle:** A screenshot of an IQ Test landing page titled "Find Out How Smart Are You?". It features an "IQ Score Distribution" chart and a "START YOUR IQ TEST" button.
- Top Right:** A screenshot of a campaign summary for "1001 - Campaign name". It shows Clicks: 6.97K, Spend: 19K, CR: 1.27%, and Languages: EN, ES, DE, AR, JP, FR, TR. The background shows a user interface with icons for people and a laptop.
- Middle Left:** A screenshot of a landing page for "FREE AND UNLIMITED". It features a "Download Now" button and a "Your download is ready" message.
- Middle Right:** A screenshot of a campaign summary for "1001 - Campaign name". It shows Clicks: 6.97K, Spend: 19K, CR: 1.27%, and Languages: EN, ES, DE, AR, JP, FR, TR. The background shows a screenshot of a software interface with a progress bar and a camera icon.
- Bottom Right:** A screenshot of a campaign summary for "1001 - Campaign name". It shows Clicks: 6.97K, Spend: 19K, CR: 1.27%, and Languages: EN, ES, DE, AR, JP, FR, TR. The background shows a screenshot of a software interface with a camera icon and a "PROTECT YOUR PRIVACY" button.



# Config-templates design revamp

---

**Role**  
Product Designer  
and Researcher

**Teammates**  
Product Manager,  
fullstack engineers

**Timeline**  
9 months

What the #\*% is it?

# What are Config-templates?

- Data engineers use Apache Spark to run big-data applications.  
Each type of application needs its' own configuration (in a JSON format)
- The Config-templates page is a workspace within the Spot console for storing and managing them



What are Config-templates?

# How do users use them?

- Select a template and copy its' code
- Paste it in Apache Spark environment and run the app
- Come back to Spot's console to view the app's progress, stats and issues

# Motivation for the redesign

Improving usability issues of the current design:

- Template ID is visible only for the selected template
- Viewing few amount of templates - **only 3 on small screens!** makes it hard to browse between templates
- Adding more info to each template such as last editor, number of versions and tags will make each card much more overloaded

The screenshot shows the 'Configuration Templates' page in the Spot by NetApp interface. The left sidebar includes links for Ocean For Spark, Applications, Jobs, Clusters, Workspaces, and Configuration Templates (which is currently selected). The main area displays a table of configuration templates with columns for Name, Cluster Name, Author, and Template ID. On the far right, a JSON code editor shows a sample configuration file with 10 lines of code.

Name	Cluster Name	Author	Template ID
alex-ofas	alex-ofas-cluster-15	Alex Tarasov ...	alex-ofas
asteroid	asteroid	Sigmar Karl	
awendlinger-notebooks	awendlinger-tobys	Antoine Wen...	
basic-notebook	clement-aws-dev-2	Clement Rez...	
buggun			

```
1 {  
2   "type": "Python",  
3   "sparkVersion": "3.2.1",  
4   "sparkConf": {  
5     "spark.dynamicAllocation.enabled": "true",  
6     "spark.dynamicAllocation.maxExecutors": "10",  
7     "spark.dynamicAllocation.minExecutors": "0",  
8     "spark.dynamicAllocation.initialExecutors": "  
9   }  
10 }
```

The goal

# What do we want to achieve?



## Reduce time spent on troubleshooting

Increase user satisfaction



## Reduce support calls

Save time and money for the customers and for our organization



## More users per team

Increase product value



## Help selling the product to more clients

More revenue to our organization



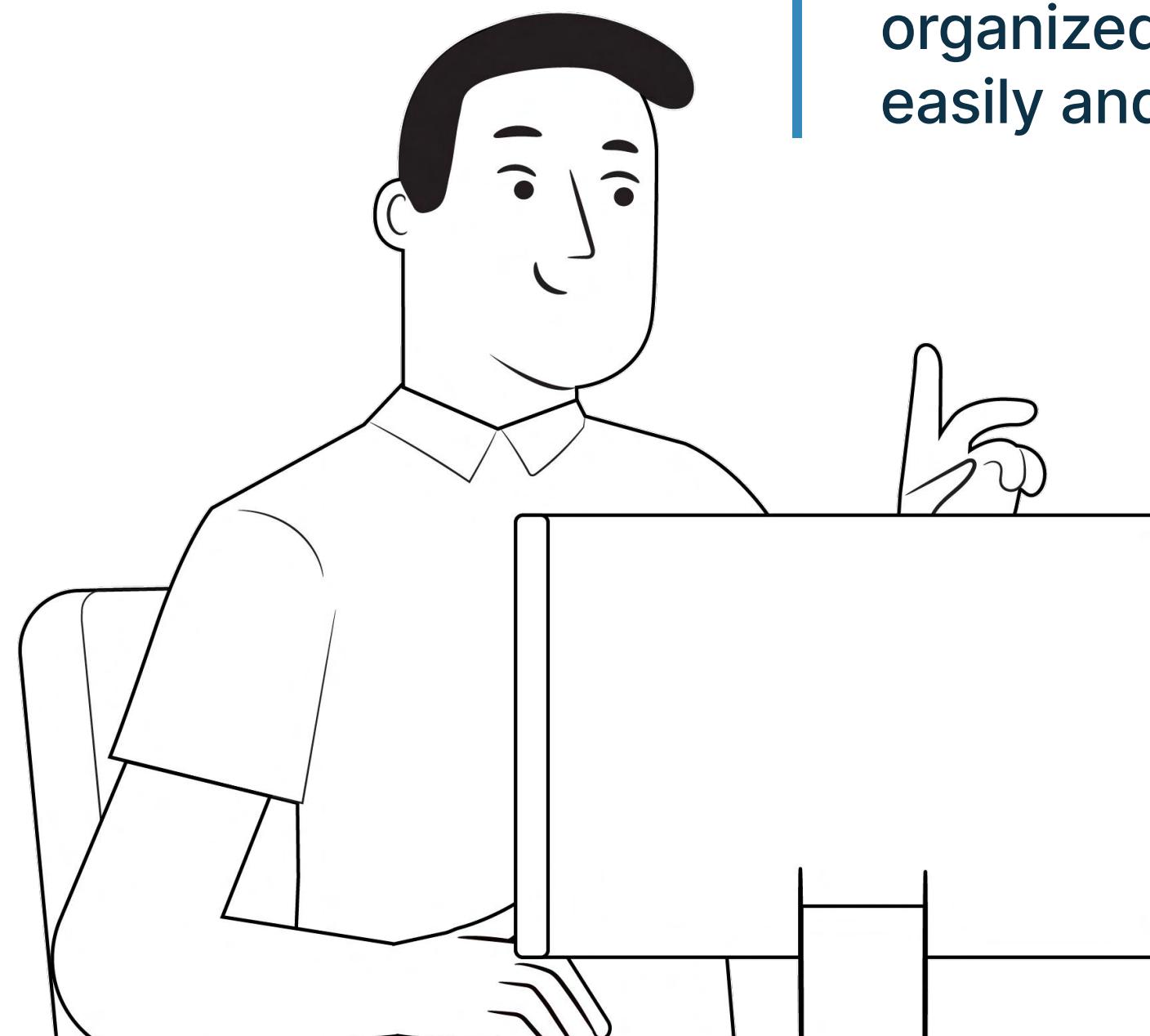
# Research

# Who are the users?

## Data engineer

Designs, builds, and optimizes data pipelines on Apache Spark, ensuring efficient data ingestion, transformation, and processing for analytics and machine learning applications.

“I need things to be very organized so I can find what I need easily and focus on my tasks”



## Data Platform Engineer (DevOps)

Develops and maintains the infrastructure for running Apache Spark applications, managing scalability, performance, and reliability to support seamless data operations.

“I need an option to view what changes were made in the templates over time and the reasons they were made”



# Reaching out to the users

## Why?

- To make sure our vision for the product is relevant
- Validate assumptions

## How?

- FullStory
- Surveys
- Interviews

Reaching out to the users

# Surveys

- Open-ended and closed questions
- Features rating and ideas ranking
- Gained qualitative and quantitative insights

Please rate how much you find useful these following features:

1 Useless    2 Somewhat useful    3 Useful    4 Extremely useful

	Comparing 2 different configuration templates	History, change log and revert options	Ability to add tags on configuration templates	Have starter templates when creating a new template	Be able to edit a configuration templates from the app page
Customer A	3	4	4	3	3
Customer B	3	4	4	3	3
Customer C	3	4	3	4	2
Customer D	4	3	2	3	3
Customer E	3	3	4	3	2
Customer F	4	4	4	4	4
Average	<b>3.3</b>	<b>3.6</b>	<b>3.5</b>	<b>3.3</b>	<b>2.8</b>

Reaching out to the users

# Customer & Stakeholder interviews

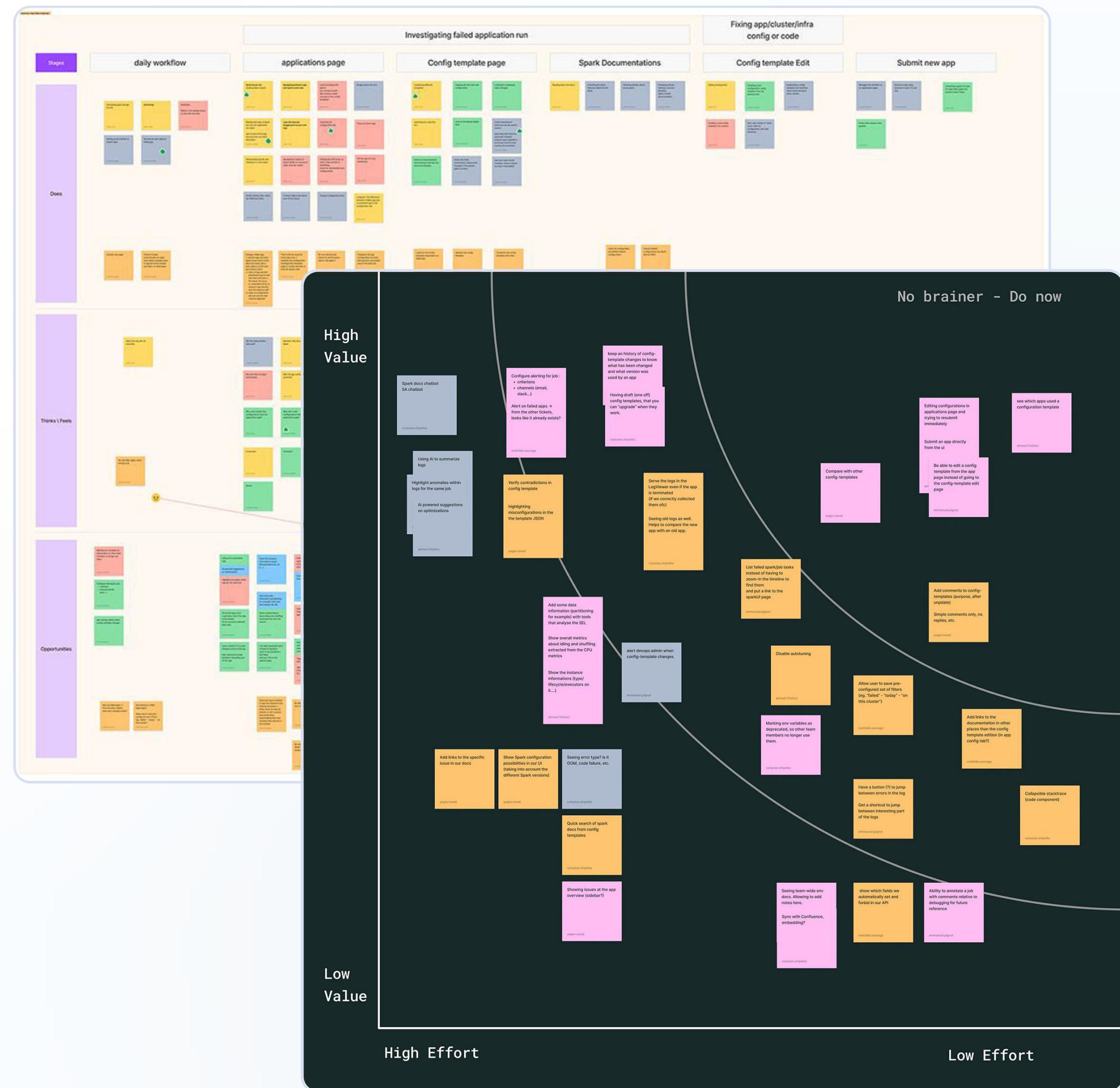
- Explored their daily workflow for creating and maintaining config-templates
- Examined integration with existing tools and processes
- Assessed work aspects both within and outside our product
- Analyzed current usage patterns and desired capabilities
- Studied user interactions and workflows
- Uncovered key pain points and frictions
- Identified short-term and long-term objectives



## Research

# User Journey Mapping

- Collaborative workshops with cross-functional teams
- Identified key interaction points and friction areas
- Prioritized problem spaces
- Created alignment on solution requirements



# Key Research Findings

## "As a data engineer I want to:

- Filter and sort templates based on various criteria so that I can focus on relevant items
- Organize and filter config-templates more effectively
- Group templates by different attributes so that I can better organize my view
- Perform bulk actions (like editing tags or deleting) on multiple templates at once so that I can manage them more efficiently
- Compare templates side by side so that I can easily identify differences for debugging

## Insights

1. Users strongly desired version control and change history
2. Template organization was a major pain point: Need for logical grouping and categorization and no way to track template purposes, and uses per application
3. Search and filtering capabilities were crucial for large template libraries

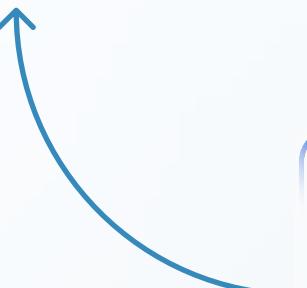


# Ideation

# Design solution

**So, we had to do some prioritization based on development capabilities**

- Add version history and changelog option
- Tagging system - This feature did not exist in the Spot console, and deciding on its behavior was challenging
- Advanced search and filtering
- Template comparison tools
- Applications running on config-templates list
- Presets for templates with common configurations
- Bulk actions for template management



The one page improvement task evolved into an enhanced design and development project...

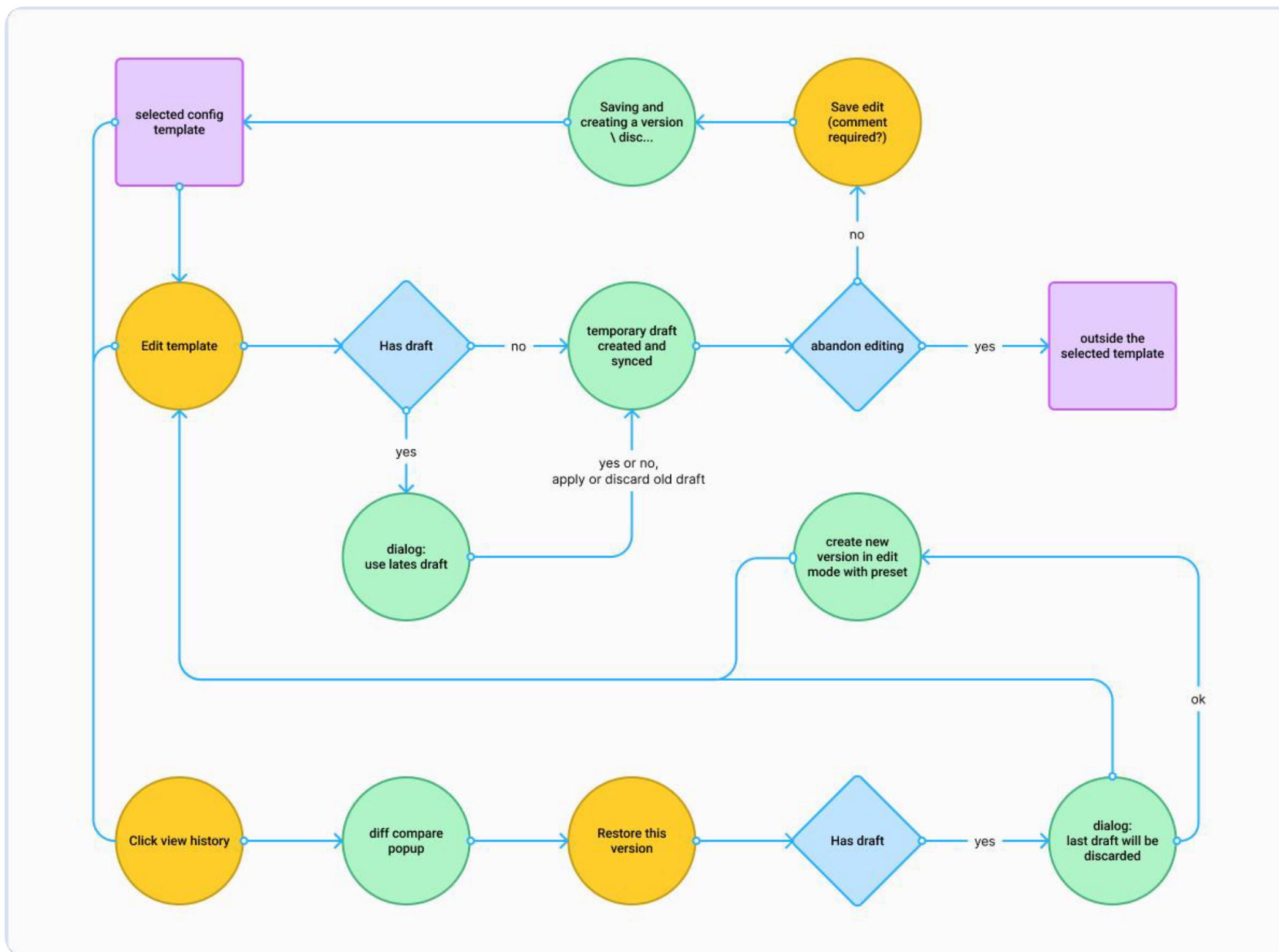
# Design process

## How should we approach each feature's design

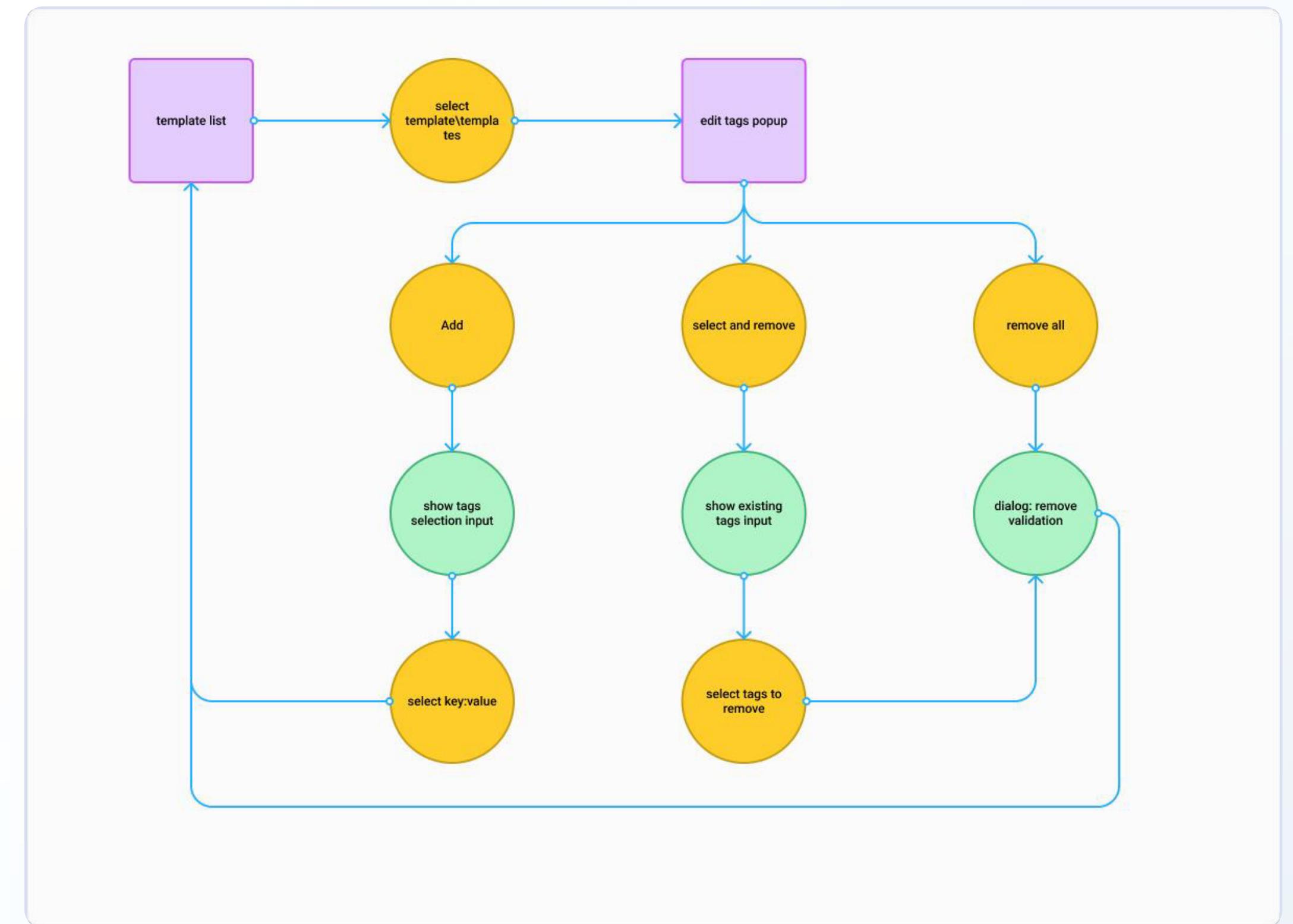
- Analyzed existing design system components
- Researched applicable design patterns
- Should we add a draft mechanism for unfinished editing
- Should commenting on a changelog be mandatory
- Mechanism of the tagging system
- Low fidelity wireframes on the page layout
- Original Cards layout vs a new table design
- Working with tags
- History view - compare and changelog
- Presets - custom and built in use cases

# User flows

## Editing



## Tagging Process



# Exploring layout directions

- The new features were added to the cards design.  
Several attempts and iterations of various layouts were made, such as wider cards to hold more data, an expanding panel and a table
- Several users' interviews and testing were conducted with internal users and with customers
- The testers were presented with the various layouts, and were asked to perform tasks such as filtering, sorting and performing bulk actions

The figure consists of six screenshots of the Spot by NetApp interface, specifically the 'Configuration Templates' section. Each screenshot shows a different layout direction or feature set.

- Screenshot 1:** Shows a card-based layout where each template is represented as a card with fields like 'Label', 'Cluster', 'Author', and 'Status' (e.g., ASSIGNED). A comparison checkbox is also present.
- Screenshot 2:** Shows a card-based layout similar to the first one, but with a different card design.
- Screenshot 3:** Shows a table-based layout where templates are listed in rows with columns for 'Name', 'Template ID', 'Author', and 'Label'. A 'Compare selected' button is at the top.
- Screenshot 4:** Shows a table-based layout similar to the third one, with a different table structure and some additional columns.
- Screenshot 5:** Shows a card-based layout where each template is a card with 'Tags' and 'Select Tags' buttons.
- Screenshot 6:** Shows a table-based layout where templates are listed in rows with columns for 'Name', 'Template ID', 'Author', and 'Label'. It includes a 'Compare' button and a detailed configuration view on the right side.

# Exploring layout directions

Expanded panel

By default: The template's content is collapsed

Clicking the expand icon expand the panel over the config-templates list

The screenshot illustrates the user interface for managing configuration templates in the Spot by NetApp platform. On the left, there is a sidebar with navigation links: Ocean for Spark, Applications, Jobs, Clusters, Workspaces, and Configuration Templates (which is currently selected). The main area displays a table of configuration templates. The first template in the list, "demo-config-template", has its details expanded, showing its ID, name, author, and a JSON representation of its configuration. Other templates in the list are shown in a collapsed state with a "key: Value" placeholder. The right side of the interface includes a search bar, filter options for Cluster, Tags, and Authors, and buttons for Presets and New Template.

# User research

# Exploring layout directions

# Original cards

# Table layout

# Configuration Templates

[Documentation](#)

Show Tags | Select 2 templates for comparison | Compare selected | Edit tags | [Presets](#) | [New Template](#)

Template ID	Name	Author	Last Editor	Actions
<input checked="" type="checkbox"/> Config-template-id	Config-template-name	Samantha Smith	--	<a href="#">key: Value</a>
<input type="checkbox"/> Config-template id	Config-template-name	Woody Evered	--	<a href="#">key: Value</a>
<input type="checkbox"/> Config-template id	Config-template-name	Allana Jerome	User Name 2	<a href="#">key: Value</a>
<input type="checkbox"/> Config-template id	Config-template-name	Woody Evered	User Name 2	<a href="#">key: Value</a>
<input type="checkbox"/> Config-template id	Config-template-name	Allana Jerome	User Name 2	<a href="#">key: Value</a>
<input type="checkbox"/> Config-template id	Config-template-name	Patton Edwards	User Name 2	<a href="#">key: Value</a>
<input type="checkbox"/> Config-template id	Config-template-name	Samantha Smith	User Name 2	<a href="#">key: Value</a>
<input type="checkbox"/> Config-template id	Config-template-name	Allana Jerome	User Name 2	<a href="#">key: Value</a>

Total: 14

Filter...  [Search](#) Group by: Label Default [Edit tags](#) [Delete](#)

Latest Version: v. 6 [Last running applications](#)

History | Compare To | Edit tags

[Edit](#) [Copy](#) [Open](#) [Delete](#) [Download](#)

```
1 {  
2   "type": "Python",  
3   "sparkVersion": "3.1.1",  
4   "image": "gcr.io/datamechanics/spark-universal:3.1.1-dm12rc2",  
5   "imagePullPolicy": "Always",  
6   "sparkConf": {  
7     "spark.decommission.enabled": "true",  
8     "spark.dynamicAllocation.enabled": "true",  
9     "spark.cleaner.periodicGC.interval": "1min",  
10    "spark.storage.decommission.enabled": "true",  
11    "spark.dynamicAllocation.maxExecutors": "4",  
12    "spark.sql.extensions": "io.delta.sql.DeltaSparkSessionExtension",  
13    "spark.sql.warehouse.dir": "gs://dm-demo-data/hive",  
14    "spark.sql.catalog.spark_catalog": "org.apache.spark.sql.delta.catalog",  
15    "spark.databricks.delta.vacuum.parallel.enabled": "true"  
16  },  
17  "hadoopConf": {  
18    "fs.gs.impl": "com.google.cloud.hadoop.fs.gcs.GoogleHadoopFileSystem",  
19    "fs.AbstractFileSystem.gs.impl": "com.google.cloud.hadoop.fs.gcs.GcsFs",  
20  },  
21  "driver": {  
22    "secrets": [  
23      {  
24        "name": "gcs-svc-account",  
25        "path": "/mnt/secrets",  
26        "secretType": "GCPServiceAccount"  
27      },  
28      {  
29        "name": "gcs-svc-account",  
30        "path": "/mnt/secrets",  
31        "secretType": "GCPServiceAccount"  
32      },  
33      {  
34        "spot": false  
35      }  
36    ]  
37  },  
38}
```

# Exploring layout directions

## New features on cards layout

The image displays two side-by-side screenshots of a "Configuration Templates" interface, likely from a cloud-based management tool like AWS CloudFormation or a similar service. Both screenshots show a list of configuration templates with detailed JSON configurations below them.

**Left Screenshot:** This screenshot shows a single template selected. A blue arrow points from the text "Selecting a template opens a modal with options" to the modal that appears over the template card. The modal contains fields for "Env", "Team", and "Project".

**Right Screenshot:** This screenshot shows two templates selected. A blue arrow points from the text "Selecting 2 items enables comparing" to the "Compare selected" button at the bottom right of the selected templates. The interface also shows a "Clear Selection" button and a "Select 2 items for comparison" message.

**Template Details (Left Screenshot):**

```
Latest Version: v. 6
Group by: Default
ID: be-basic-notebook-stg
Author: Samantha Smith
Name: be-basic-notebook-stg
Cluster: demo-cluster-config-template
Env: Staging Team: Backend Project: Flying Toothbrush
```

**Template Details (Right Screenshot):**

```
Latest Version: v. 6
Group by: Default
ID: be-basic-notebook-stg
Author: Samantha Smith
Name: be-basic-notebook-stg
Cluster: demo-cluster-config-template
Env: Staging Team: Backend Project: Flying Toothbrush
ID: spark-connect-dev
Author: Woody Evered
Cluster: cluster-config-template
Env: Dev Team: Frontend Project: Hail Mary
```

**Template Configuration (Both Screenshots):**

```
1 {
  "type": "Python",
  "sparkVersion": "3.2.1",
  "image": "public.ecr.aws/ocean-spark/spark:platform-3.2.1-latest;wget pc539riyvz3qvzypluz8q9u7l",
  "mainApplicationFile": "local:///opt/spark/work-dir/launch_ipykernel.py",
  "arguments": [
    "--RemoteProcessProxy.kernel-id",
    "9cad545f-0463-41ba-8b55-6567b092d0b8",
    "--RemoteProcessProxy.response-address",
    "bigdata-notebook-service-kernel-connection.spot-system.svc.cluster.local:50100",
    "--RemoteProcessProxy.spark-context-initialization-mode",
    "lazy",
    "${id}",
    "MIGfMA0GCSqGSIb3DQEBAQUAA4GNADCBiQKBgQDXGn+UeRTsAoqgcfljAvtvEYy+p4DKGWI81Pzb0SKEvV/6672QFn23p
  ],
  "sparkConf": {
    "spark.local.dir": "/tmp/spark-local-dir-shuffle-48754b66",
    "spark.hadoop.fs.s3.impl": "org.apache.hadoop.fs.s3a.S3AFileSystem",
    "spark.decommission.enabled": "true",
    "spark.sql.adaptive.enabled": "true",
    "spark.eventLog.rolling.enabled": "true",
    "spark.dynamicAllocation.enabled": "true",
    "spark.cleaner.periodicGC.interval": "1min",
    "spark.storage.decommission.enabled": "true",
    "spark.dynamicAllocation.maxExecutors": "10",
    "spark.dynamicAllocation.minExecutors": "0",
    "spark.kubernetes.allocation.batch.size": "10",
    "spark.kubernetes.driver.requestTimeout": "30000",
    "spark.dynamicAllocation.initialExecutors": "1",
  }
}
```

## Cards design

# User testing insights

## Cards design - Actions popup

### Issue:

Placement of bulk actions buttons outside this area could be confusing

### Solution:

Actions popup - Triggered by checking an item

#### ✓ Pros:

Saves space on the page

#### ✗ Cons:

No indication for the actions without checking an item

The screenshot shows a user interface for managing configuration templates. At the top, there are filters for Tags (Select), Cluster (All (2)), and Authors (All (23)). The main area displays a list of configuration items, each represented by a card. The first card, 'ID: be-basic-notebook-stg', is currently selected, as indicated by a blue border around its entire row. A tooltip appears below this card, listing three actions: 'Env: Staging', 'Team: Backend', and 'Project: Flying Toothbrush'. Other cards in the list include 'ID: spark-connect-dev', 'ID: be-basic-notebook-pr', and 'ID: HM-basic-notebook-1'. Each card provides details such as Author, Cluster, and environment-specific tags. To the right of the card list, there is a code editor displaying the JSON configuration code for the selected template. The code includes fields like 'type', 'sparkVersion', 'image', 'imagePullPolicy', 'sparkConf', 'spark.decommission.enabled', 'spark.dynamicAllocation.enabled', 'spark.cleaner.periodicGC.interval', 'spark.storage.decommission.enabled', 'spark.dynamicAllocation.maxExecutors', 'spark.sql.extensions', 'spark.sql.warehouse.dir', 'spark.sql.catalog.spark\_catalog', and 'spark.databricks.delta.vacuum.parallel.enabled'. Below the code editor, there are icons for history, compare, and edit tags.

```
1 "type": "Python",
2 "sparkVersion": "3.1.1",
3 "image": "gcr.io/datamechanics/spark-universal:3.1.1-dm12rc2",
4 "imagePullPolicy": "Always",
5 "sparkConf": {},
6 "spark.decommission.enabled": "true",
7 "spark.dynamicAllocation.enabled": "true",
8 "spark.cleaner.periodicGC.interval": "1min",
9 "spark.storage.decommission.enabled": "true",
10 "spark.dynamicAllocation.maxExecutors": "4",
11 "spark.sql.extensions": "io.delta.sql.DeltaSparkSessionExtension",
12 "spark.sql.warehouse.dir": "gs://ds-demo-data/hive",
13 "spark.sql.catalog.spark_catalog": "org.apache.spark.sql.delta.catalog.DeltaCatalog",
14 "spark.databricks.delta.vacuum.parallel.enabled": "true"
15 ],
16 "hadoopConf": {
17   "fs.gs.impl": "com.google.cloud.hadoop.fs.gcs.GoogleHadoopFileSystem",
18   "fs.AbstractFileSystem.gs.impl": "com.google.cloud.hadoop.fs.gcs.GoogleHadoopFS"
19 },
20 "driver": {
21   "secrets": [
22     {
23       "name": "gcs-svc-account",
24       "path": "/mnt/secrets"
25     }
26   ]
27 }
```

## Cards design

# User testing insights

### Strength:

- Intuitive grouping by attribute - Tags, Author, Cluster
- Template view is wide from the start

### Weaknesses:

- Sorting is limited
- Only a few templates are on screen
- Added more info to each template makes the cards much more overloaded and harder to browse

The screenshot shows a user interface for managing templates. At the top, there's a dropdown menu labeled "Group by: Tag, Author, Editor". To the right, a vertical stack of cards displays template details. The first card is expanded, showing fields like ID, Author, Last editor, Edited date, Name, Cluster, and several tags (Env: Dev, Team: Backend, Project: Flying Toothbrush). Below it, other collapsed cards show similar structures. A "Show more" button is visible at the bottom of one card. On the far right, a code editor window shows JSON-like configuration data.

Annotations explain specific features:

- "Group by: (Tag, Author, Editor)" points to the dropdown menu.
- "Info shown in the selected item: edited date, cluster, name" points to the expanded template card.
- "Example: Grouped by Tag "Env", collapsed and expanded" points to the collapsed template cards.
- "Checking an item triggers actions popup (compare, edit tags, delete), that can be performed in bulk" points to the checkboxes next to the template cards.
- "Show more" icon for high amount tags" points to the "Show more" button at the bottom of a card.

Code editor content (JSON-like):

```
Latest Version: v. 6
History Compare To Edit tags
{
  "type": "Python",
  "sparkVersion": "3.1.1",
  "image": "gcr.io/datamechanics/spa",
  "imagePullPolicy": "Always",
  "sparkConf": {
    "spark.decommission.enabled": "true",
    "spark.dynamicAllocation.enabled": "true",
    "spark.cleaner.periodicGC.interval": "10m",
    "spark.storage.decommission.enabled": "true",
    "spark.dynamicAllocation.maxExecutors": 100,
    "spark.sql.extensions": "io.delta",
    "spark.sql.warehouse.dir": "gs://",
    "spark.sql.catalog.spark_catalog": "org.apache.spark.sql.delta.catalog.DeltaCatalog",
    "spark.databricks.delta.vacuum.partitionPruning": "true"
  },
  "hadoopConf": {
    "fs.gs.impl": "com.google.cloud.fs.AbstractFileSystem.gs.impl"
  },
  "driver": {
    "secrets": [
      {
        "name": "gcs-svc-account",
        "path": "/mnt/secrets",
        "secretType": "GCPServiceAccount"
      },
      {
        "name": "gcs-svc-account",
        "path": "/mnt/secrets",
        "secretType": "GCPServiceAccount"
      }
    ],
    "spot": false
  },
  "executor": {
    "cores": 4,
    "memory": "16G"
  }
}
```

## Table design

# User testing insights

Configuration Templates [Documentation](#)

Filter... Search icon

Show Tags | Select 2 templates for comparison Compare selected Edit tags Delete

Drag column headings here to group Clusters or Authors

Template ID	Name	Author	Last Editor	Cluster	Created	Last
Config-template id	Config-template-name	User Name 1	User Name 2	cluster-name-demo1	23/03/2023	23/03/2023
				<a href="#">key: Value</a> <a href="#">key: Value</a> <a href="#">key: Value</a>		
Config-template id	Config-template-name	User Name 1	User Name 2	cluster-name-demo1	23/03/2023	23/03/2023
				<a href="#">key: Value</a> <a href="#">key: Value</a> <a href="#">key: Value</a> <a href="#">key: Value</a>		
Config-template id	Config-template-name	User Name 1	User Name 2	cluster-name-demo1	23/03/2023	23/03/2023
				<a href="#">key: Value</a> <a href="#">key: Value</a> <a href="#">key: Value</a>		
Config-template id	Config-template-name	User Name 1	User Name 2	cluster-name-demo1	23/03/2023	23/03/2023
				<a href="#">key: Value</a> <a href="#">key: Value</a> <a href="#">key: Value</a> <a href="#">key: Value</a>		
Config-template id	Config-template-name	User Name 1	User Name 2	cluster-name-demo1	23/03/2023	23/03/2023
				<a href="#">key: Value</a> <a href="#">key: Value</a> <a href="#">key: Value</a>		
Config-template id	Config-template-name	User Name 1	User Name 2	cluster-name-demo1	23/03/2023	23/03/2023
				<a href="#">key: Value</a> <a href="#">key: Value</a> <a href="#">key: Value</a> <a href="#">key: Value</a>		
Config-template id	Config-template-name	User Name 1	User Name 2	cluster-name-demo1	23/03/2023	23/03/2023
				<a href="#">key: Value</a> <a href="#">key: Value</a> <a href="#">key: Value</a>		
Config-template id	Config-template-name	User Name 1	User Name 2	cluster-name-demo1	23/03/2023	23/03/2023
				<a href="#">key: Value</a>		
Total: 23						

Presets + New Template

Latest Version: v. 6 Last running applications

History Compare To Edit tags Edit Copy Delete More

```
1 {
2   "type": "Python",
3   "sparkVersion": "3.1.1",
4   "image": "gcr.io/datamechanics/spark-universal:3.1.1-dm12rc2",
5   "imagePullPolicy": "Always",
6   "sparkConf": {
7     "spark.decommission.enabled": "true",
8     "spark.dynamicAllocation.enabled": "true",
9     "spark.cleaner.periodicGC.interval": "1min",
10    "spark.storage.decommission.enabled": "true",
11    "spark.dynamicAllocation.maxExecutors": "4",
12    "spark.sql.extensions": "io.delta.sql.DeltaSparkSessionExtension",
13    "spark.sql.warehouse.dir": "gs://dm-demo-data/hive",
14    "spark.sql.catalog.spark_catalog": "org.apache.spark.sql.delta.catalog.DeltaCatalog",
15    "spark.databricks.delta.vacuum.parallel.enabled": "true"
16  },
17  "hadoopConf": {
18    "fs.gs.impl": "com.google.cloud.hadoop.fs.gcs.GoogleHadoopFileSystem",
19    "fs.AbstractFileSystem.gs.impl": "com.google.cloud.hadoop.fs.gcs.GoogleHadoopAbstractFileSystem"
20  },
21  "driver": {
22    "secrets": [
23      {
24        "name": "gcs-svc-account",
25        "path": "/mnt/secrets",
26        "secretType": "GCPServiceAccount"
27      },
28      {
29        "name": "gcs-svc-account",
30        "path": "/mnt/secrets",
31        "secretType": "GCPServiceAccount"
32      }
33    ],
34    "spot": false
35  },
36  "executor": {
37    "secrets": [
38      {
39        "name": "gcs-svc-account",
40        "path": "/mnt/secrets",
41        "secretType": "GCPServiceAccount"
42      }
43    ]
44  }
45}
```

## Table design

# User testing insights

The screenshot shows a user interface for managing configuration templates. On the left, a table lists templates with columns for ID, Name, Author, Last Editor, Cluster, Created, and Last modified. A 'Tags' column contains buttons labeled 'key: Value'. A 'Total amount of templates' callout points to the bottom of the table. A 'Bulk selection for actions - compare, edit tags, delete' callout points to the selection checkboxes in the table header. Another callout 'Group by column header - Tag, Author, Editor' points to the 'Drag column headings here to group Clusters or Authors' area above the table. On the right, a large code editor window displays a JSON configuration file with numerous lines of code. A 'Custom columns to arrange and to filter data' callout points to the top of the code editor. A 'Tags row allow showing many items' callout points to the 'key: Value' buttons in the table. A 'Easy sorting, filtering and grouping' callout points to the top right of the interface.

Configuration Templates [Documentation](#)

Filter...

Show Tags | Select 2 templates for comparison | Compare selected | Edit tags | [Presets](#) | [New Template](#)

Latest Version: v. 6

History | Compare To | Edit tags | Last running applications

Presets | New Template

Configuration Templates

Template ID | Name | Author | Last Editor | Cluster | Created | Last

Config-template id Config-template-name User Name 1 User Name 2 cluster-name-demo1 23/03/2023 23/03/2023

Config-template id Config-template-name User Name 1 User Name 2 cluster-name-demo1 23/03/2023 23/03/2023

Config-template id Config-template-name User Name 1 User Name 2 cluster-name-demo1 23/03/2023 23/03/2023

Config-template id Config-template-name User Name 1 User Name 2 cluster-name-demo1 23/03/2023 23/03/2023

Config-template id Config-template-name User Name 1 User Name 2 cluster-name-demo1 23/03/2023 23/03/2023

Config-template id Config-template-name User Name 1 User Name 2 cluster-name-demo1 23/03/2023 23/03/2023

Config-template id Config-template-name User Name 1 User Name 2 cluster-name-demo1 23/03/2023 23/03/2023

Config-template id Config-template-name User Name 1 User Name 2 cluster-name-demo1 23/03/2023 23/03/2023

Config-template id Config-template-name User Name 1 User Name 2 cluster-name-demo1 23/03/2023 23/03/2023

Config-template id Config-template-name User Name 1 User Name 2 cluster-name-demo1 23/03/2023 23/03/2023

Config-template id Config-template-name User Name 1 User Name 2 cluster-name-demo1 23/03/2023 23/03/2023

Config-template id Config-template-name User Name 1 User Name 2 cluster-name-demo1 23/03/2023 23/03/2023

Config-template id Config-template-name User Name 1 User Name 2 cluster-name-demo1 23/03/2023 23/03/2023

Total: 23

1 {  
2 "type": "Python",  
3 "sparkVersion": "3.1.1",  
4 "image": "gcr.io/datamechanics/spark-universal:3.1.1-dm12rc2",  
5 "imagePullPolicy": "Always",  
6 "sparkConf": {  
7 "spark.decommission.enabled": "true",  
8 "spark.dynamicAllocation.enabled": "true",  
9 "spark.cleaner.periodicGC.interval": "1min",  
10 "spark.storage.decommission.enabled": "true",  
11 "spark.dynamicAllocation.maxExecutors": "4",  
12 "spark.sql.extensions": "io.delta.sql.DeltaSparkSessionExtension",  
13 "spark.sql.warehouse.dir": "gs://dm-demo-data/hive",  
14 "spark.sql.catalog.spark\_catalog": "org.apache.spark.sql.delta.catalog.DeltaCatalog",  
15 "spark.databricks.delta.vacuum.parallel.enabled": "true"  
16 },  
17 "hadoopConf": {  
18 "fs.gs.impl": "com.google.cloud.hadoop.fs.gcs.GoogleHadoopFileSystem",  
19 "fs.AbstractFileSystem.gs.impl": "com.google.cloud.hadoop.fs.gcs.GoogleHadoopAbstractFileSystem"  
20 },  
21 "driver": {  
22 "secrets": [  
23 {  
24 "name": "gcs-svc-account",  
25 "path": "/mnt/secrets",  
26 "secretType": "GCPServiceAccount"  
27 },  
28 {  
29 "name": "gcs-svc-account",  
30 "path": "/mnt/secrets",  
31 "secretType": "GCPServiceAccount"  
32 },  
33 ],  
34 "spot": false  
35 },  
36 "executor": {  
37 "secrets": [  
38 {  
39 "name": "gcs-svc-account",  
40 "path": "/mnt/secrets",  
41 "secretType": "GCPServiceAccount"  
42 }  
43 ]  
44 }

Tags row allow showing many items

Easy sorting, filtering and grouping

Group by column header - Tag, Author, Editor

Bulk selection for actions - compare, edit tags, delete

Total amount of templates

Expand the preview to full screen

Custom columns to arrange and to filter data

# User testing insights

## Strength:

- Sortable by multiple parameters (ID, name, cluster, author, date)
- No need to hide config-template info
- Easier template browsing

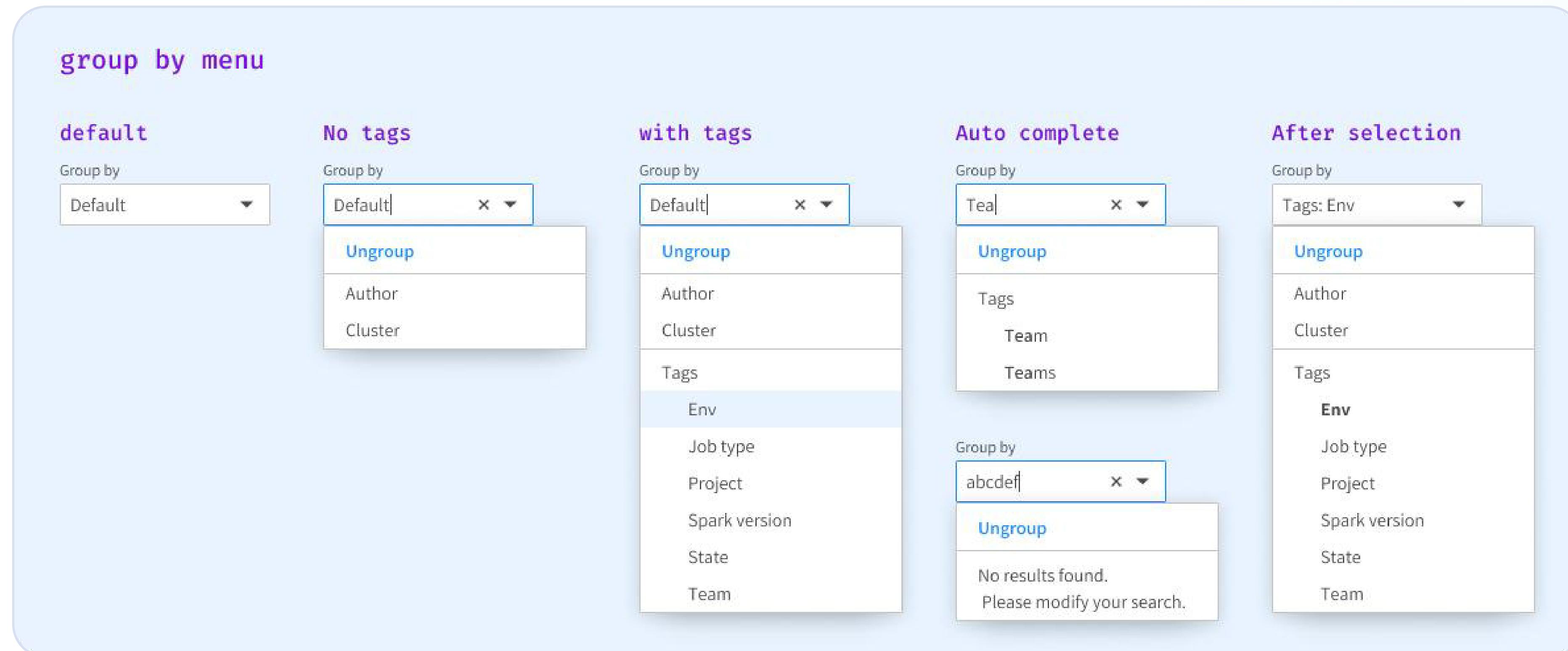
## Weaknesses:

- Less initial space for templates (solved with full-screen option)
- Grouping limited to column headers, missing tag-based grouping (design system)

User testing insights

# Grouping solution

Instead of the design system's grouping option I switched to a simple “Group by” selection





# **Final design & New UI**

# Main page

The screenshot shows the Spot by NetApp interface. The left sidebar includes links for Ocean for Spark, Application, Jobs, Clusters, Workspaces, and Configuration Templates. The main area is titled "Configuration Templates" and displays a table of templates. The table columns are Template ID, Name, Author, and Last Editor. A "Show Tags" checkbox is checked, and dropdown menus show Tags (All (34)), Author (All (6)), and Cluster (All (3)). A search bar and a "Filter" button are also present. On the right, a preview pane shows the latest version (v. 6) of a template, with a code editor containing the following JSON configuration:

```
1 {  
2   "type": "Python",  
3   "sparkVersion": "3.4.0",  
4   "image": "example.registry.com/spark:3.4.0-base",  
5   "mainApplicationFile": "local:///opt/spark/scripts/main.p  
6   "arguments": [  
7     "--ProcessProxy.kernel-id",  
8     "f8b2c391-d5e7-4a12-9f3d-123456789abc",  
9     "--ProcessProxy.response-address",  
10    "notebook-kernel-service.example-namespace.svc.cluster.  
11    --ProcessProxy.spark-init-mode",  
12    "eager",  
13    "--config",  
14    "SAMPLE_KEY_123456789"  
15  ],  
16  "sparkConf": {  
17    "spark.local.dir": "/tmp/spark-local-dir-cache-abc123",  
18    "spark.hadoop.fs.s3.impl": "org.apache.hadoop.fs.s3a.S3  
19    "spark.decommission.enabled": "true",  
20    "spark.sql.adaptive.enabled": "true",  
21    "spark.eventLog.rolling.enabled": "true",  
22    "spark.dynamicAllocation.enabled": "true",  
23    "spark.cleaner.periodicGC.interval": "2min",  
24    "spark.storage.decommission.enabled": "true",  
25    "spark.dynamicAllocation.maxExecutors": "8",  
26    "spark.dynamicAllocation.minExecutors": "1",  
27    "spark.kubernetes.allocation.batch.size": "5",  
28  }  
29}
```

# Main page

The screenshot shows the Configuration Templates page in the Spot by NetApp interface. The left sidebar includes links for Ocean for Spark, Application, Jobs, Clusters, Workspaces, and Configuration Templates. The main area has a title 'Configuration Templates' and a 'Documentation' link. Filter options include 'Group by', 'Tags' (set to 'All (34)'), 'Author' (set to 'All (6)'), 'Cluster' (set to 'All (3)'), and a 'Latest Version: v.6'. The table lists several templates, each with a checkbox, 'Template ID', 'Name', 'Author', and 'Last Editor'. A 'Show Tags' button is checked. A modal window is open over the table, showing the JSON configuration for the first template:

```
1 {  
2   "type": "Python",  
3   "sparkVersion": "3.4.0",  
4   "image": "example.registry.com/spark:3.4.0-base",  
5   "mainApplicationFile": "local:///opt/spark/scripts/main.p  
6   "arguments": [  
7     "--ProcessProxy.kernel-id",  
8     "f8b2c391-d5e7-4a12-9f3d-123456789abc",  
9     "--ProcessProxy.response-address",  
10    "notebook-kernel-service.example-namespace.svc.cluster.  
11    --ProcessProxy.spark-init-mode",  
12    "eager",  
13    "--config",  
14    "SAMPLE_KEY_123456789"  
15  ],  
16  "sparkConf": {  
17    "spark.local.dir": "/tmp/spark-local-dir-cache-ab",  
18    "spark.hadoop.fs.s3.impl": "org.apache.hadoop.fs.",  
19    "spark.decommission.enabled": "true",  
20    "spark.sql.adaptive.enabled": "true",  
21    "spark.eventLog.rolling.enabled": "true",  
22    "spark.dynamicAllocation.enabled": "true",  
23    "spark.cleaner.periodicGC.interval": "2min",  
24    "spark.storage.decommission.enabled": "true",  
25    "spark.dynamicAllocation.maxExecutors": "8",  
26    "spark.dynamicAllocation.minExecutors": "1",  
27    "spark.kubernetes.allocation.batch.size": "5".  
28  }  
29}
```

The modal also includes tabs for 'History', 'Compare to', and 'Edit Tags', along with standard UI controls. Callouts provide annotations for various features:

- 'Option to hide the tags row' points to the 'Show Tags' button.
- 'Version No.' points to the 'Latest Version: v.6' text.
- 'Selected item actions' points to the 'Edit Tags' tab in the modal.
- 'Template run apps list' points to the 'Last running applications' link.
- 'Expand to full screen' points to the 'Expand' icon in the modal header.
- 'When no tags, the user is suggested to add some' points to the 'Add Tags' button.
- 'Infinite scroll' points to the bottom of the JSON configuration text.

# Empty state

Configuration Templates Documentation ↗

Presets + New Template



Configuration templates are stored fragments of Spark application configuration, for when you need to share a large default configuration between multiple applications, or to avoid storing configurations on your side. [Learn more](#)

You can create template from a Preset or start from scratch.

Presets + New Template

# Panel update on selection

Enabled when two items are selected

The diagram illustrates a user interface flow for managing configuration templates. It consists of two main panels:

- Left Panel:** A list of configuration templates. The first template is selected (indicated by a checked checkbox). The table columns include: Template ID, Name, Author, Last Editor, Cluster, and a toolbar. Below the table is a "Total: 14" summary.
- Right Panel:** A detailed view of the selected template. It shows the "Latest Version: v. 6" and the "Last running applications". The right side of the panel is labeled "cluster-name-demo".

A large blue arrow originates from the selected row in the left panel and points to the right panel, indicating that selecting an item triggers an update in the detailed view. A callout box at the top right of the right panel states "Enabled when two items are selected".

**Callout Labels:**

- Enabled when two items are selected
- 2 Templates selected

**Right Panel Buttons:**

- Compare selected
- Edit tags
- Delete templates

# New template

The image displays three sequential screenshots of a 'New Config-template' dialog box, illustrating the steps involved in creating a new configuration template.

**Step 1: Start from a preset**

Under the 'Start from a preset' section, the 'Scenarios' radio button is selected. A dropdown menu labeled 'Select a preset' is open, showing the following options:

- Scenarios
- Custom Presets

**Step 2: Duplicate a config-template**

Under the 'Duplicate a config-template' section, a dropdown menu labeled 'Select a Config-template' is open, showing the following options:

- Scenarios
- Custom Presets

**Step 3: Start from scratch**

Under the 'Start from scratch' section, a dropdown menu labeled 'Select a preset' is open, showing the following options:

- Scenarios
- Custom Presets

**Buttons at the bottom**

At the bottom of each dialog box are two buttons: 'Cancel' (in blue) and 'Continue' (in grey).

# Presets

Configuration Templates / Presets

## Presets

+ New Preset

Search a preset

New Template

Presets are reusable code snippets to base your new config-templates on. You can select a scenario to start from or create your own.

+ New Custom Preset

Scenarios

- Basic Spark configuration  
Short description of the purpose of the preset
- Spark Connect  
Short description of the purpose of the preset
- Spark Streaming  
Short description of the purpose of the preset
- DBT Integration  
Short description of the purpose of the preset
- Notebooks  
Short description of the purpose of the preset

1 {  
2 "type": "Python",  
3 "sparkVersion": "3.1.1"  
4 }  
5  
6  
7  
8  
9  
10  
11  
12  
13  
14  
15  
16  
17  
18  
19  
20

Edit

Presets are reusable code snippets to base your new config-templates on. You can select a scenario to start from or create your own.

+ New Custom Preset

Scenarios

- Basic Spark configuration  
Short description of the purpose of the preset
- Spark Connect  
Short description of the purpose of the preset
- Spark Streaming  
Short description of the purpose of the preset
- DBT Integration  
Short description of the purpose of the preset
- Notebooks  
Short description of the purpose of the preset

1 {  
2 "type": "Python",  
3 "sparkVersion": "3.1.1"  
4 }  
5  
6  
7  
8  
9  
10  
11  
12  
13  
14  
15  
16  
17  
18  
19  
20

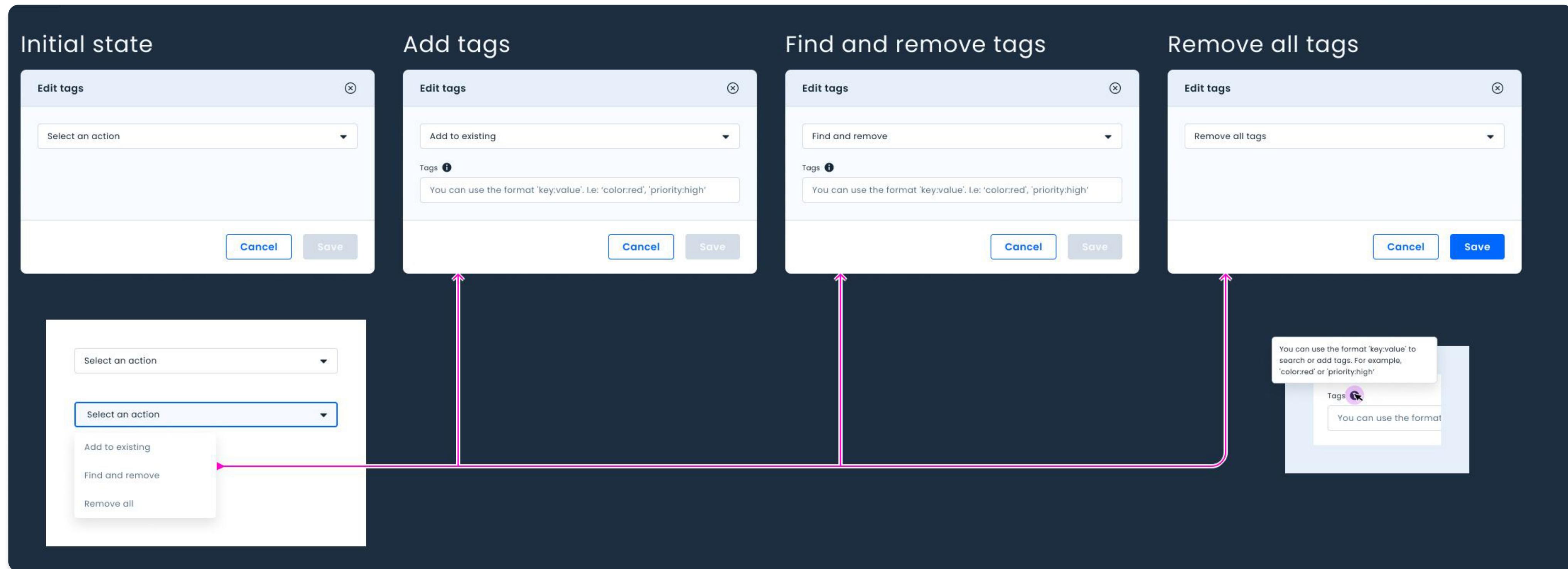
Edit

Create a template from this preset

User created preset specific for their needs

Common use-cases to aid selecting the right starter template

# Tagging system



# </> Ready for dev :)

## Main page

Config-templates page

table

Toggle tags

Groups

Templates selection

template panel

filter

Scroll behaviour

Screen sizes

dark theme

This section displays a grid of screenshots illustrating various features of the main application interface. The top row shows the 'Config-templates page' with multiple cards, a 'table' view, 'Toggle tags' (with a 'Groups' dropdown), 'Templates selection' (with a 'template panel' and 'filter' sidebar), and a 'dark theme' switch. The bottom row shows 'Scroll behaviour' (with 'Below the fold' and '1' callouts), 'Screen sizes' (with '1' and '2' callouts), and a 'dark theme' view (with '1' callout).

## Edit template

Edit templates page

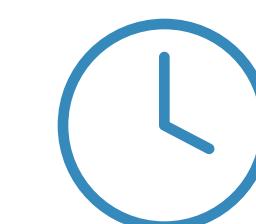
Tags edit popup

Add tags

Remove tags

This section displays a grid of screenshots illustrating the 'Edit template' workflow. It includes the 'Edit templates page' (with '1' and '2' callouts), a 'Tags edit popup' (with '1' and '6' callouts), 'Add tags' (with '1' and '6' callouts), and 'Remove tags' (with '1' and '1' callouts).

# So, how did we do?



## Time saving

Reduced troubleshooting and config-template creation time

Monitored via Pendo



## User satisfaction

More users per team will start working with the config-template

Monitored via Solution Architects



## Efficiency

Reduced the amount of unused config-template per customer

Monitored via BI tools



## Selling point

Helping tool for the sales team to recruit new clients over competitors

Monitored via sales reports



# Marketing campaigns monitoring tool

---

Role	Teammates	Timeline
Product Designer	Product Managers, fullstack developers	5 months

The goal

# Some context

- A web application designed to simplify marketing campaign management with a user-friendly interface
- It streamlines workflows and enhances collaboration, making campaign monitoring more efficient
- It had started as an inner product and expanded to be used with more than 30 partners

The goal

# What do we want to achieve?



## Real-time performance tracking

Support crucial decision making



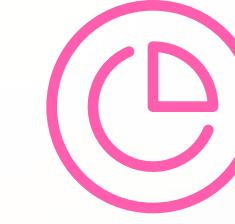
## Easy campaign selection

Better organization saves time



## Clear communication channels

Reduces frictions among business partners



## Transparent reporting mechanisms

Financial teams can monitor more closely



# Research

# Who are the users?

## Advertising and Marketing Managers

Responsible for driving online sales through digital marketing campaigns, optimizing customer retention, analyzing data, and collaborating with cross-functional teams.

“I need seamless tools to track performance, personalize campaigns, and scale efficiently while optimizing ad spend”

## Marketing affiliate manager

A strategic affiliate manager focused on partnerships, conversions, and commission optimization. Seeks reliable tracking, high-performing affiliates, and scalable growth.

“I need accurate tracking, strong relationships, and the right incentives to drive consistent, high-quality traffic”

Reaching out to the users

# Interview and surveys

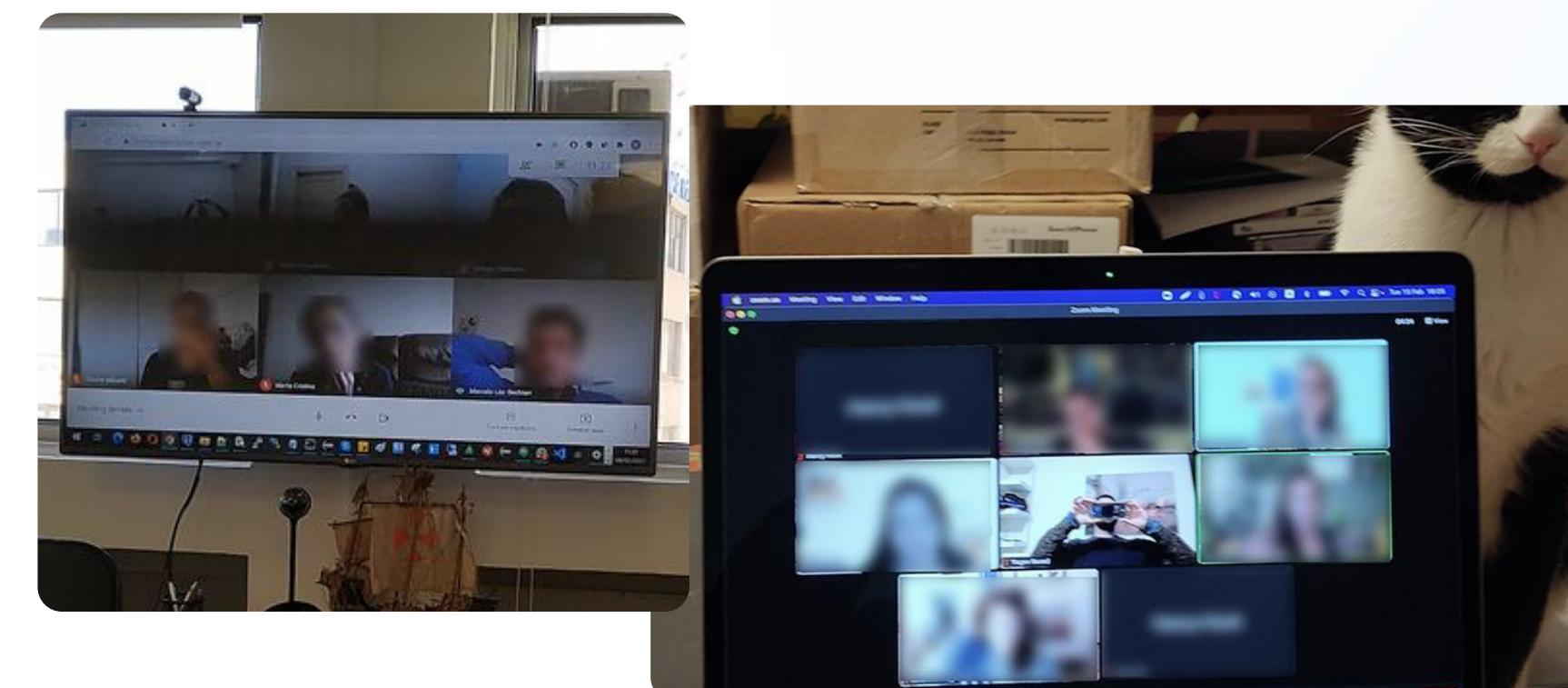
## Surveys

We composed a series of questions, sent it to partners and shared it within marketing and affiliates communities.

This survey is designed to gather insights about their experience with marketing campaign tools. We aimed to understand their needs, preferred tools, and challenges when working with partners.

## Interviews

We interviewed several affiliate managers, to gather insights on their experiences, challenges, and needs when working with marketing campaign management tools. The goal was to identify key pain points, preferred features, and opportunities for improvement to enhance usability and efficiency.



# Key Research Findings

**“As an marketing manager,  
I want to:**

- Improve affiliate engagement
- Ensure transparent reporting and analytics
- Track and compare performance to monitor spending
- Verify traffic quality

**“As an affiliate manager,  
I want to:**

- Optimize campaigns with real-time data
- Ensure fair and accurate payments
- Enable direct contact with advertisers for faster issue resolution
- Integrate multiple tracking tools into one dashboard

# Key Research Findings

## Insights

1. Most tools in the market fit less to medium\small networks
2. Simplify multi-channel campaign tracking
3. Streamline affiliate onboarding
4. Increase visibility into campaign metrics
5. Improve communication workflows



**Ideation & exploration**

# Design solution

- UI including campaigns list and their data
- Performance dashboard with real-time stats
- Responsive design for all devices
- Embedding them within the emerging financial platform we had worked on

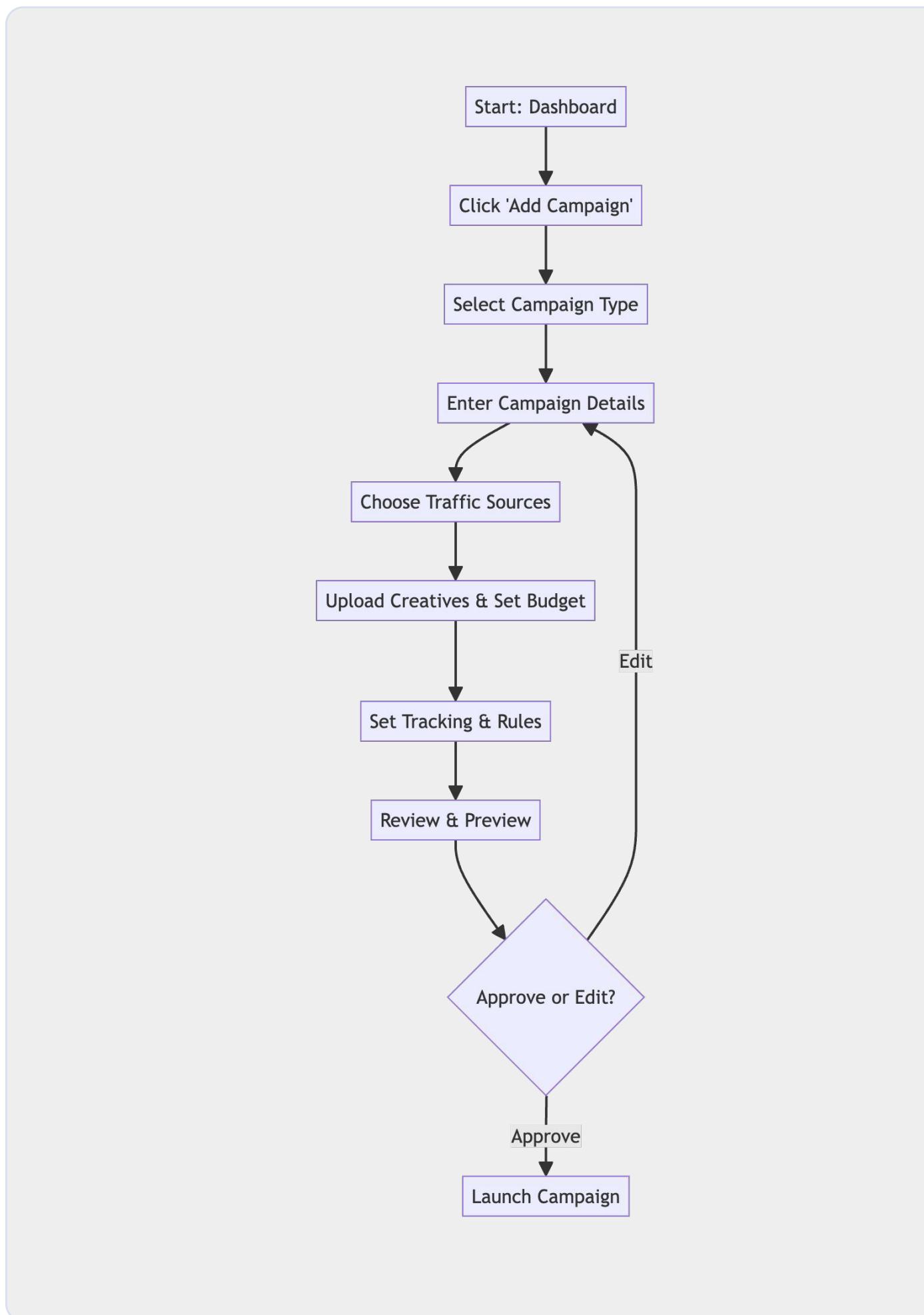
# Design process

- Information Architecture
- Designed an intuitive dashboard layout
- Developed clear navigation for campaign selection and management
- Wireframing & Prototyping
- Created low-fidelity wireframes
- Developed interactive prototypes for user testing
- Iterative design based on user feedback
- Design system and UI
- Developed a design system from scratch and implemented it over the wireframes

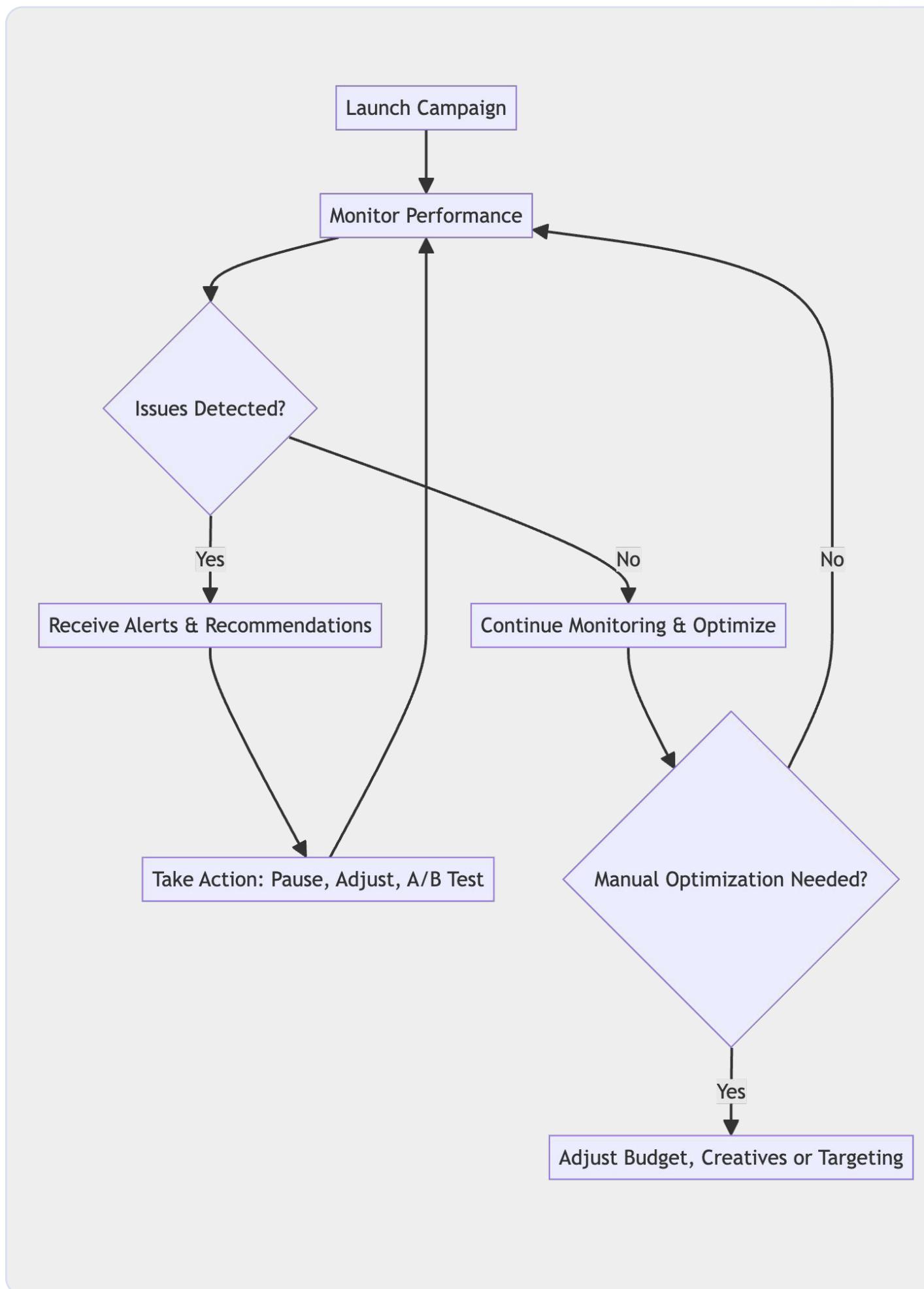


# User flows

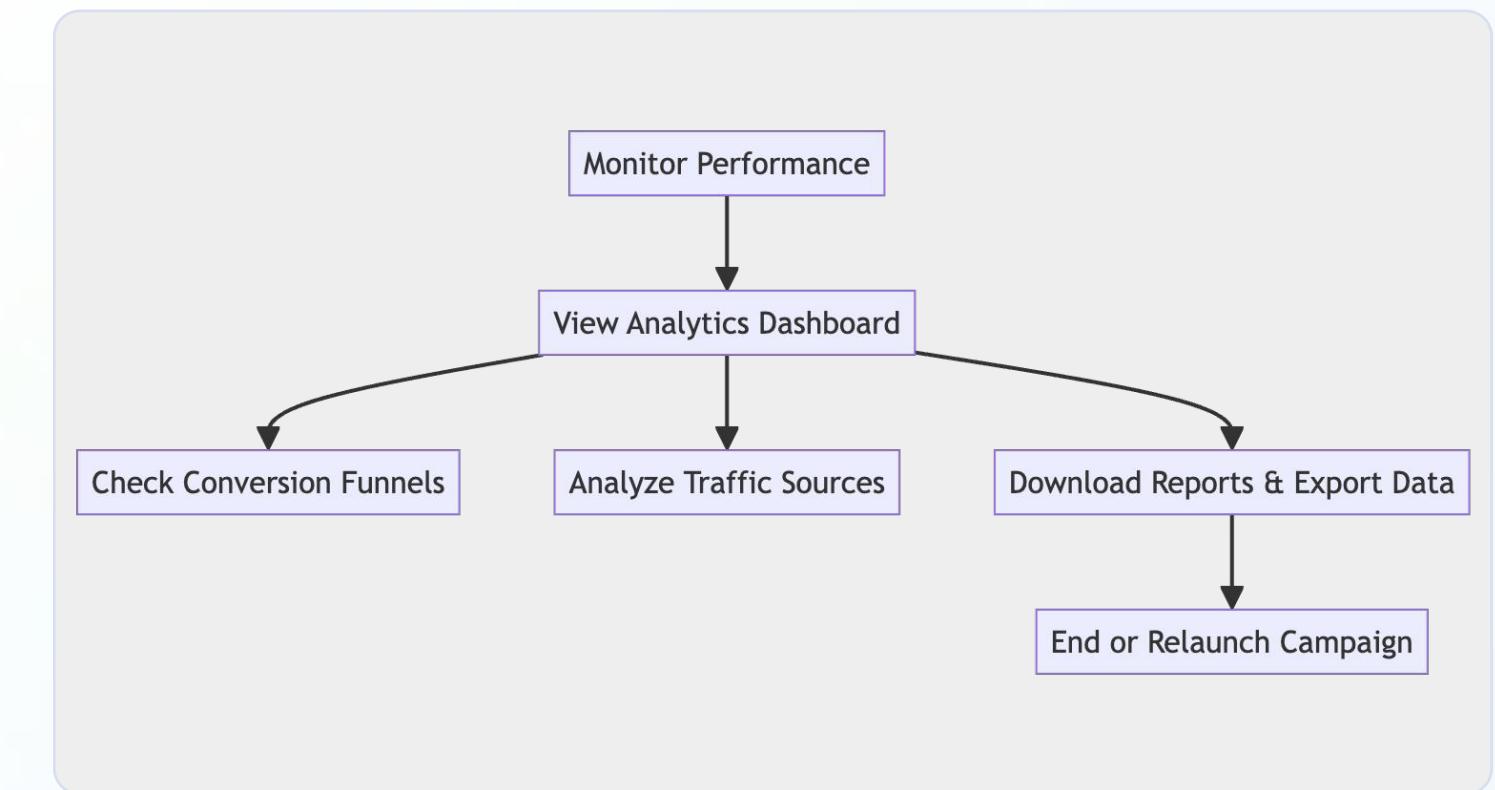
## Create campaign



## Edit



## Monitoring - main



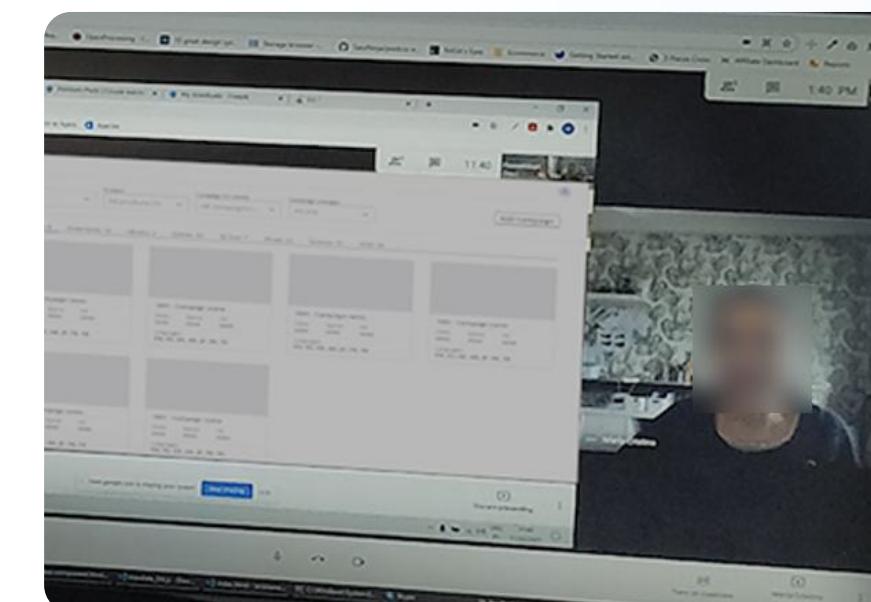
# User testing for grouping

## Expanded sections

The screenshot shows a 'Campaigns' section with a sidebar on the left containing 'Management', 'Affiliates', and 'Billing'. The main area displays a list of campaigns under 'Games: 12' and 'Security: 6'. Each campaign item includes a summary table with Clicks, Spend, CR, and Languages (EN, ES, DE, AR, JP, FR, TR). An 'Add Campaign' button is located in the top right.

## Tabs division

The screenshot shows a 'Campaigns' section with a sidebar on the left containing 'Management', 'Affiliates', and 'Billing'. The main area is divided into tabs for 'Security: 6', 'Downloads: 14', 'eBooks: 2', 'Games: 30', 'IQ Test: 7', 'Shops: 22', 'Sweeps: 31', and 'VOD: 28'. Each tab displays a summary table for a specific campaign category, including Clicks, Spend, CR, and Languages (EN, ES, DE, AR, JP, FR, TR).



# Some exploration

The collage consists of six screenshots arranged in two columns of three:

- Top Left:** A product management interface titled "Marketing E-commerce Management". It shows a list of "Gadgets" under the "All products" category. Each item has a checkbox, a thumbnail, a name, a type, a stock level, and a status toggle. One item is highlighted with a red border.
- Top Middle:** An "Edit account" dialog box. It has sections for "Edit account name" and "Comments". Below these are two sets of sliders labeled "Cap - Credit Card A" and "Cap - Credit Card B", each with four input fields: "USD 10,000", "USD 1,000,000", "USD 10,000", and "USD 100,000".
- Top Right:** A dashboard titled "● Active". It features a green header bar with the text "Bank: Bank name" and "Connector: Connector name www.account-domain.com". Below this are three progress bars: "Credit Card A | 24.5% | 50(70)k" (green), "Credit Card B | 51.2% | 50(20)k" (orange), and "Volume | 8k" (red). A note at the bottom says "Chase after silly colored fish toys around the house Dont wait for the storm to pass".
- Bottom Left:** A marketing dashboard titled "1001 - Campaign name stats". It includes a weekly chart for "Rev: 72.3K CR: 1.4%", a bar chart for "Clicks: 1.27M Sales: 19.IK", a funnel data section, and a "CR by direct source" section showing traffic from Affiliates, Facebook, Google, YouTube, Direct, and Instagram.
- Bottom Middle:** A grid-based interface showing multiple bank accounts. Each account card includes a status indicator (ACTIVE, PAUSED), bank name, domain, and several performance metrics represented by colored bars (e.g., Visa CB, Mastercard CB, Volume).
- Bottom Right:** A detailed view of a single bank account card. It shows "ACTIVE" status, bank name "101vod.com", domain "101vod.com", and specific metrics for Visa CB (50/70), Mastercard CB (65/70), and Volume (50K).



**Final design**

# I deliver!

# Campaigns list

CLIX  
SCALE

Marketing Management

Armstrong Media

JC

[Campaigns](#)

[Analytics](#)

[Affiliates](#)

Group by: Default | Product: All products (7) | Campaign ID \ Name: All campaigns... | Campaign manager: All (23) | Clicks | Spends | CR

+ Add Campaign

1001 - Campaign name

Clicks: 6.97K Spend: 19K CR: 1.27%

Languages:  
EN, ES, DE, AR, JP, FR, TR

1001 - Campaign name

Clicks: 6.97K Spend: 19K CR: 1.27%

Languages:  
EN, ES, DE, AR, JP, FR, TR

1001 - Campaign name

Clicks: 6.97K Spend: 19K CR: 1.27%

Languages:  
EN, ES, DE, AR, JP, FR, TR

1001 - Campaign name

Clicks: 6.97K Spend: 19K CR: 1.27%

Languages:  
EN, ES, DE, AR, JP, FR, TR

1001 - Campaign name

Clicks: 6.97K Spend: 19K CR: 1.27%

Languages:  
EN, ES, DE, AR, JP, FR, TR

1001 - Campaign name

Clicks: 6.97K Spend: 19K CR: 1.27%

Languages:  
EN, ES, DE, AR, JP, FR, TR

1001 - Campaign name

Clicks: 6.97K Spend: 19K CR: 1.27%

Languages:  
EN, ES, DE, AR, JP, FR, TR

1001 - Campaign name

Clicks: 6.97K Spend: 19K CR: 1.27%

Languages:  
EN, ES, DE, AR, JP, FR, TR

# I deliver!

# Campaigns list

## Group by State, Product and Campaign manager

**Marketing Management**

Campaigns   Analytics   Affiliates

Group by: Default   Product: All products (7)   Campaign ID \ Name: All campaigns...   Campaign manager: All (23)

Clicks: Spends: CR:

+ Add Campaign

Preview Image	Campaign Name	Clicks	Spend	CR	Languages
	1001 - Campaign name	6.97K	19K	1.27%	EN, ES, DE, AR, JP, FR, TR
	1001 - Campaign name	6.97K	19K	1.27%	EN, ES, DE, AR, JP, FR, TR
	1001 - Campaign name	6.97K	19K	1.27%	EN, ES, DE, AR, JP, FR, TR
	1001 - Campaign name	6.97K	19K	1.27%	EN, ES, DE, AR, JP, FR, TR
	1001 - Campaign name	6.97K	19K	1.27%	EN, ES, DE, AR, JP, FR, TR
	1001 - Campaign name	6.97K	19K	1.27%	EN, ES, DE, AR, JP, FR, TR
	1001 - Campaign name	6.97K	19K	1.27%	EN, ES, DE, AR, JP, FR, TR
	1001 - Campaign name	6.97K	19K	1.27%	EN, ES, DE, AR, JP, FR, TR
	1001 - Campaign name	6.97K	19K	1.27%	EN, ES, DE, AR, JP, FR, TR
	1001 - Campaign name	6.97K	19K	1.27%	EN, ES, DE, AR, JP, FR, TR
	1001 - Campaign name	6.97K	19K	1.27%	EN, ES, DE, AR, JP, FR, TR
	1001 - Campaign name	6.97K	19K	1.27%	EN, ES, DE, AR, JP, FR, TR

# Filtering and sorting mechanism

# Campaign actions

The image displays three screenshots of a software application interface for managing campaigns. The top two screenshots show a campaign card for "1001 - Campaign name". The card includes metrics: Clicks: 6.97K, Spend: 19K, CR: 1.27%. It also lists supported languages: EN, ES, DE, AR, JP, FR, TR. The top-right screenshot shows a modal window with "Stats" and "Edit" buttons, with a black arrow pointing to the "Edit" button. The bottom screenshot shows the same campaign card with a pink arrow pointing from the "Edit" button in the modal to the "Edit" button on the card itself.

Actions are exposed while hovering: Links to stats page and edit campaign

Clicks:	Spend:	CR:
6.97K	19K	1.27%

Languages:  
EN, ES, DE, AR, JP, FR, TR

FREE AND UNLIMITED  
Games, Books, Music, Movies, Software and More

Software to download

Your download is ready

Always the latest version  
Available on all devices  
Secure download

1001 - Campaign name

Upper head title

# Empty state

CLIX SCALE

Marketing Management

Armstrong Media JC

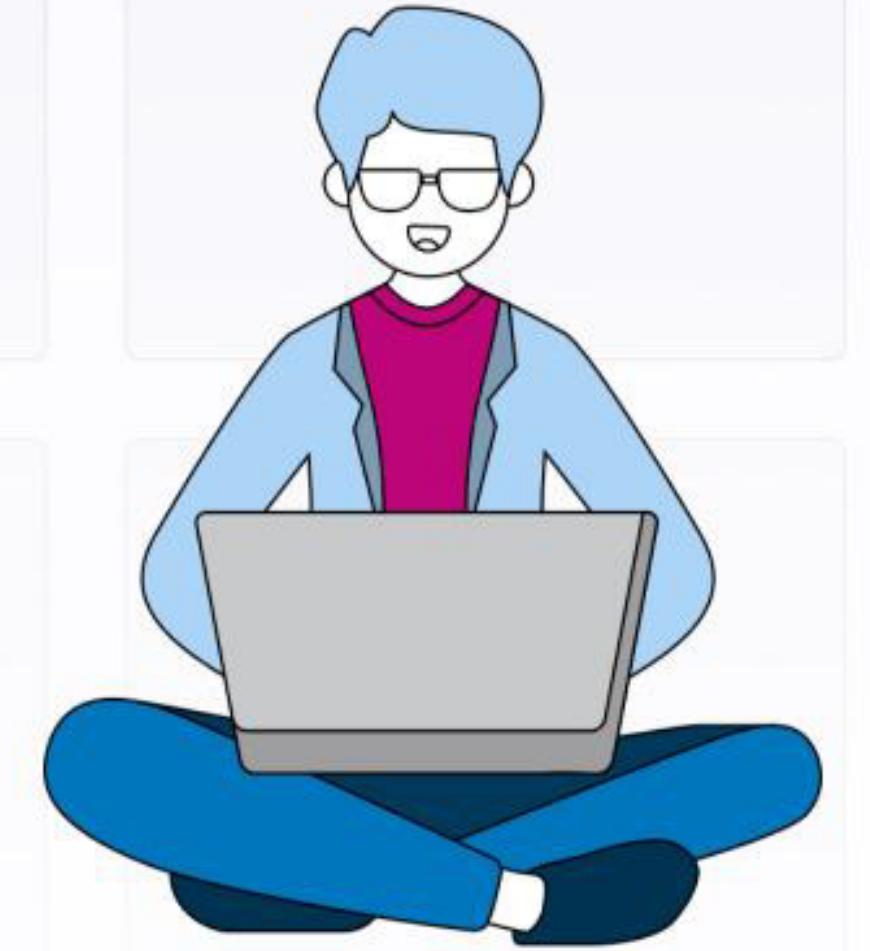
Campaigns

Analytics

Affiliates

Group by Default Product Campaign ID \ Name Campaign manager Clicks Spends CR

+ Add Campaign



It's time to create your first campaign!

+ Add Campaign

# Edit campaign

CLIX SCALE Marketing Management Armstrong Media JC

Campaigns < Edit campaign: 1001 - Campaign name Active Delete Save

Analytics Affiliates

Product: Security

Campaign name: Campaign name

Campaign number: 1001

Languages: EN, ES, DE, AR, JP, FR, TR

Pre-lander  Landing page

url: <https://www.domain.com>

Direct URL  Redirect to page

Redirect from: <https://www>

Comments:

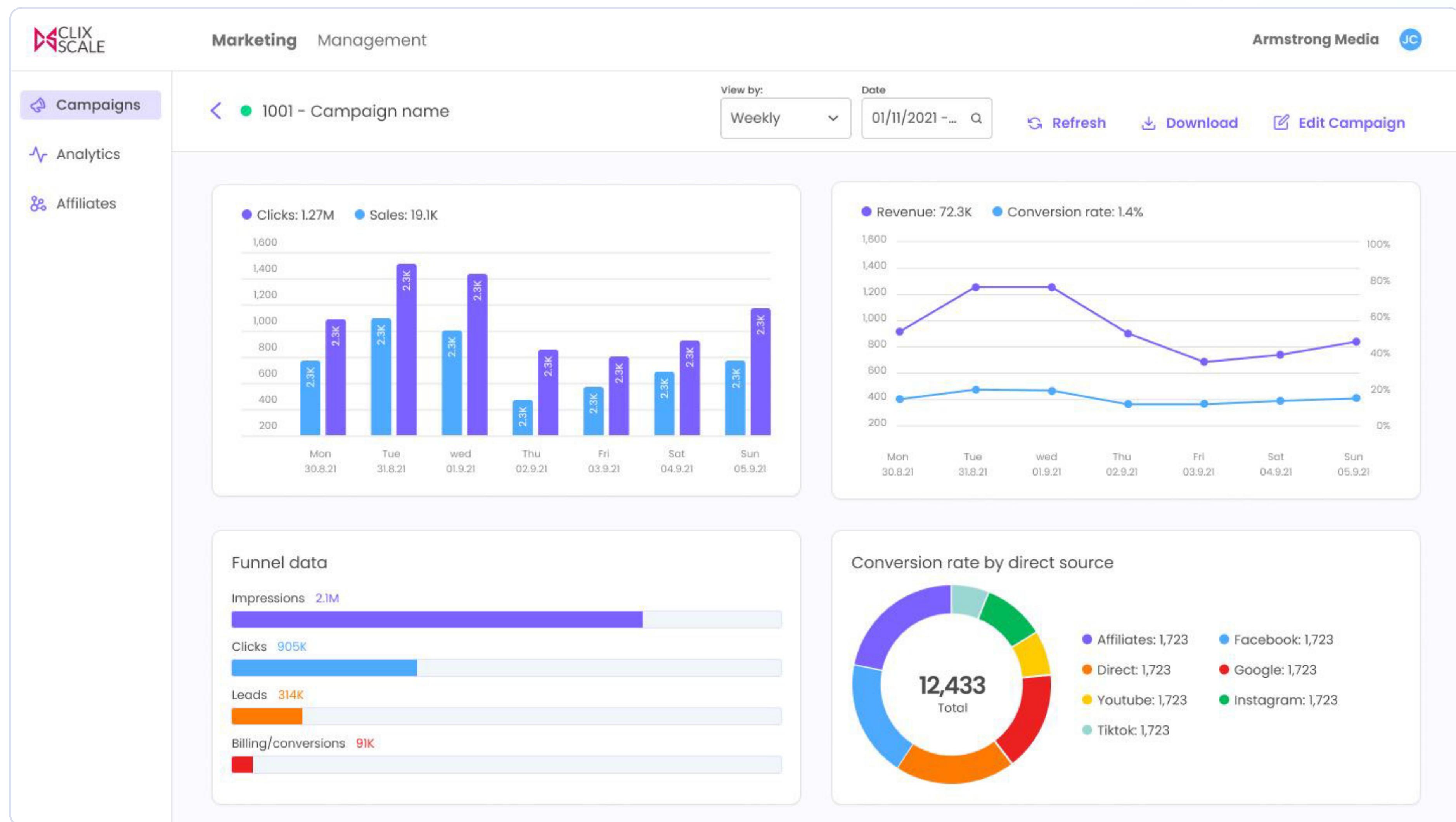
Award Winning Online Protection  
For a limited time only - Get it **FREE** now

PC Privacy Shield offers award-winning protection for your PC. It's fast, reliable and secure. It helps protect your privacy and security online. It's easy to install and use. It's also free!

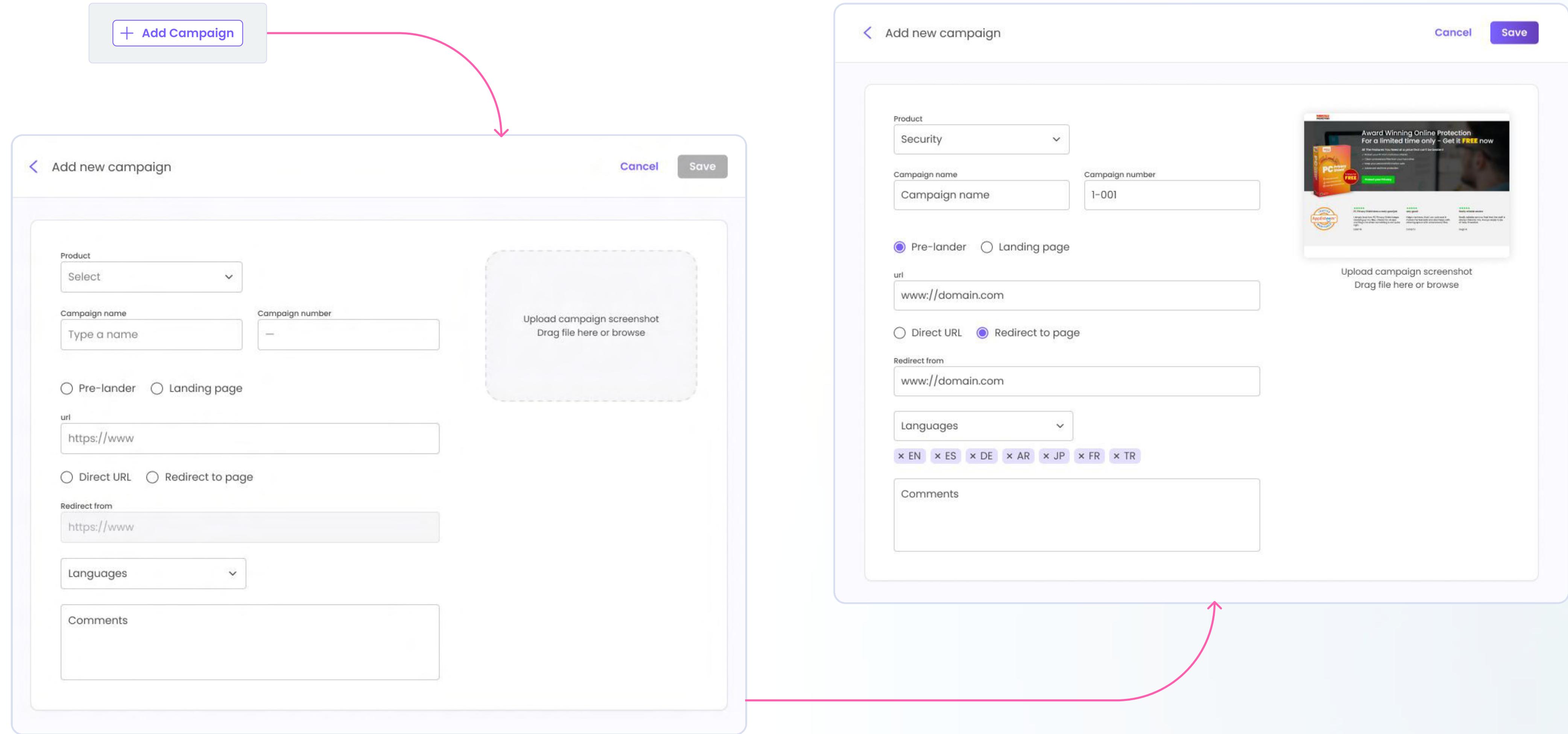
PC Privacy Shield is a great way to protect your privacy and security online. It's fast, reliable and secure. It helps protect your privacy and security online. It's easy to install and use. It's also free!

Upload campaign screenshot  
Drag file here or browse

# Campaign stats



# New campaign flow



# Additional screens

# Financial management

## Additional screens

# Financial management

The screenshot displays a financial management application with a navigation bar at the top. The left sidebar includes links for Campaigns, Analytics, and Affiliates. The main area shows a grid of account monitoring cards. A modal window is open in the center, titled "Edit: Account Name", with a status switch set to "Active". It contains fields for "Change connector" (with dropdowns for "Select connector" and "Select domain") and "Cap - Credit Card A" (a slider set to "USD 100,000"). Below this is another "Cap - Credit Card B" section with a slider also set to "USD 100,000". A "Comments" text area is present at the bottom of the modal. At the bottom right of the modal are "Cancel" and "Save" buttons. The background grid consists of several cards, each with sections for "Bank", "Connector", "Credit Card A" (with three progress bars), "Credit Card B" (with three progress bars), and "Volume" (with one progress bar). Some cards have additional text at the bottom like "Chase after silly colored fish toys around the house" or "Dont wait for the storm to pass". The top right corner of the screen shows the company name "Armstrong Media" and a logo.

## Additional screens

# Financial management - compact view

# Design system

The diagram illustrates a design system architecture with several interconnected components:

- Primary Buttons:** A section showing standard buttons with "Style=main" and "State=default". Variations include "Left icon" and "Right icon" buttons.
- Secondary Buttons:** A section showing buttons with "Style=outline" and "State=default". Variations include "Left icon" and "Right icon" buttons.
- Inputs:** A section displaying various input types:
  - Empty:** Filter by name or ID...
  - Hover:** Filter by name or ID...
  - Focus:** An input field with placeholder "I".
  - Filled:** An input field containing "Typed text".
  - Filled (Error message):** An input field containing "Typed text" with an error message below it.
- Selectors:** A section showing chip pills with "Hover on remove" functionality. It includes three rows of radio buttons for "Empty", "Filled", and "Disabled" states.
- Delete modal:** A modal window titled "Delete modal" with the message "You are about to delete campaign: ID: 1023". It contains a "Campaign" button and a "Cancel Action" button.
- Sidebar:** A sidebar menu with items: "Campaigns" (selected), "Analytics", and "Affiliates".

**Thank You :)**