ACONA Technical Concept

General Userstories:

- As a websites owner I can check one of my pages and get personalized recommendations what can be improved
- As a website owner I see how important metrics has been improved/declined in time for specific pages
- As a website owner I see what pages are working well and what pages need to be improved or have great potential.
- As a website owner I get notifications when an important metric is not behaving like forecasted (e.g. OrganicClicks suddenly drop, and it's not weekend)

Goals technically:

- Save url and time specific data in a central way that can be retrieved via an API call
- Compare success metrics to other data for a specific time

General approaches and resources:

https://medium.com/walmartlabs/how-we-build-a-robust-analytics-platform-using-spark-kafka-and-cassandra-lambda-architecture-70c2d1bc8981

Example Log Data wrangling:
“Scalable Log Analytics with Apache Spark — A comprehensive case-study” von Dipanjan (DJ) Sarkar https://link.medium.com/Gg8cGdnpZ3

Time series aggregation (do it in the stream):
https://github.com/killrweather/killrw
https://www.slideshare.net/SparkSummit/streaming-analytics-with-spark-kafka-cassandra-and-akka-by-helena-edelsoneather

ETL Pipeline:
“Building an ETL Pipeline in Python” von Daniel Foley
https://towardsdatascience.com/building-an-etl-pipeline-in-python-f96845089635

Kafka streaming:
https://databricks.com/blog/2017/04/04/real-time-end-to-end-integration-with-apache-kafka-in-apache-sparks-structured-streaming.html

Clean Data: “cleanframes — data cleansing library for Spark!” von Dawid Rutowicz
https://link.medium.com/DLRmQ9Su03

“Create your first ETL Pipeline in Apache Spark and Python” von Adnan Siddiqi
https://towardsdatascience
o.com/create-your-first-etl-pipelinek-in-apache-spark-and-python-ec3d12e2c169

Lambda architecture:
https://towardsdatascience.com/lambda-architecture-how-to-build-a-big-data-pipeline-part-1-8b56075e83fe

GraphX Spark library: https://spark.apache.org/docs/latest/graphx-programming-guide.html
Spark Intro: https://stanford.edu/~rezab/sparkclass/slides/reza_introtalk.pdf

Job scheduling

Jobs/Scheduler/Data pipeline orchestration and execution:

How often is data updated/imported?
When is it done?
Limits?
Authentication?

Sources: Matomo API, Lighthouse API, Crawler+Analyzer, Google Search Console

What is needed?

- List of clients with URLs to be analyzed.
- Endpoint (like matomo instance) and Authentication for each client
- Interval for scheduling of the job

Result of its job: Tool saves normalized page metrics and variables in the data warehouse.

Such jobs can be done regularly for a fixed set of urls or full page (crawls), but should be also possible to be initialized for a single url just in time (usecase: Update data / analyse for a specific url directly in the cms).

Technical solutions:

- Apache Airflow

Good intro: https://www.youtube.com/watch?v=YWtfU0MQZ_4
https://towardsdatascience.com/getting-started-with-apache-airflow-df1aa77d7b1b

https://towardsdatascience.com/getting-started-with-airflow-locally-and-remotely-d068df7fcb4

Example with docker-compose file using airflow, kafka, spark:
https://medium.com/@rose4earn/docker-compose-ing-kafka-airflow-spark-b2ea66993c50

Apache Airflow example:
https://towardsdatascience.com/step-by-step-build-a-data-pipeline-with-airflow-4f96854f7466
Code: https://github.com/kyokin78/airflow

Creating DAGs dynamically:
https://www.astronomer.io/guides/dynamically-generating-dags/

Create connections programmatically:
https://stackoverflow.com/questions/51863881/is-there-a-way-to-create-modify-connections-through-airflow-api

Production Docker Images for Apache Airflow:
https://www.youtube.com/watch?v=wDr3Y7q2XoI&feature=youtu.be
Example with Analytics Data and some good tipps:
https://medium.com/leboncoin-engineering-blog/data-traffic-control-with-apache-airflow-ab8fd3fc8638

DAGs:

Create DAG for every client or try to use one pipeline for one service?

1) Matomo; Scheduling: Daily; Job 1: Get current list of clients, domains and authentications from ACONA Administration Tool Job 2: Export data (tbd) for every url from Matomo and save in a postgres db. Job3: Aggregate it and save results in the Data Warehouse.
2) Google Search Console; Scheduling: Daily.
3) Lighthouse

- Spark Streaming API
- Kafka

Example using Kafka and Spark to listen on tweets/slack and post to slack channel dependent on sentiment: https://lenadroid.github.io/posts/distributed-data-streaming-action.html

ETL workflow frameworks: http://bytepawn.com/luigi-airflow-pinball.html (~~probably too much~~)

ACONA Data Warehouse

What is a Data Lake? https://databricks.com/glossary/data-lake

Storage options:

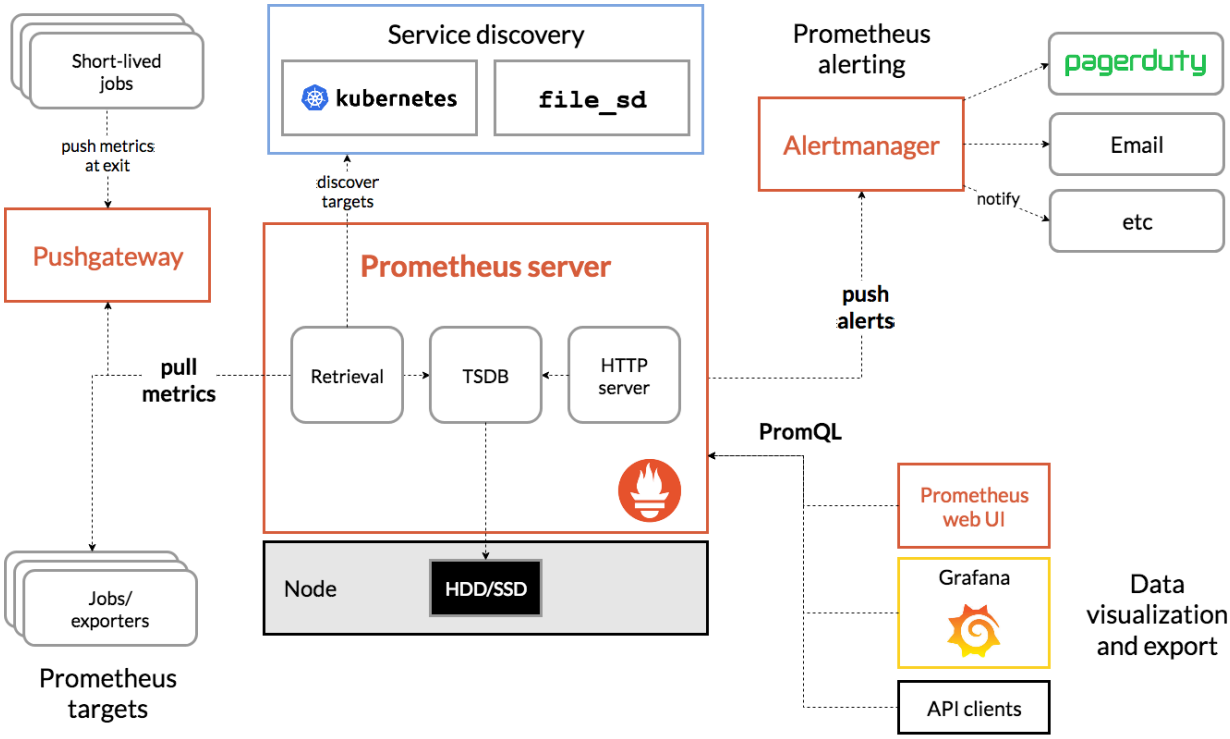
1) https://github.com/timescale/promscale Promscale is an open-source long-term store for Prometheus data designed for analytics built on TimeScaleDB (Postgres)
   Writing to PromScale:
   https://github.com/timescale/promscale/blob/master/docs/writing_to_promscale.md

2) ScyllaDB: https://www.scylladb.com/
3) TimeScaleDB: https://www.timescale.com/

Promscale and Prometheus

https://prometheus.io/docs/introduction/overview/#overview

Benefits: widely used, supports jobs, incl. Alert manager, provides multiple integration like a query API and grafana



Data model schema:
https://github.com/timescale/promscale/blob/master/docs/sql_schema.md#data-model-schema

Data exchange via OpenMetrics format:
https://github.com/OpenObservability/OpenMetrics/blob/master/specification/draft-richih-opsawg-openmetrics-01.txt

Good intro to Prometheus data model: https://sysdig.com/blog/prometheus-metrics/

For user specific grafana frontend (SaaS), this approach looks handy:
https://medium.com/@richard.holly/powerful-saas-solution-for-detection-metrics-c67b9208d362 and https://gitlab.com/optima_public/prometheus_oauth_proxy
This way we could use Drupal as an oauth server and then let users access their data in one grafana dashboard.

Example for an exporter (data source): https://github.com/teamzerolabs/covid_exporter
https://medium.com/teamzerolabs/write-aprometheus-api-exporter-in-record-time-7518ee22f238

Jupyter notebook integration: https://pypi.org/project/prometheus-pandas/

Somehow related: Monitoring system by CERN (partly using kafka and spark to aggregate data and also use prometheus and grafana):
https://arxiv.org/pdf/2007.03630.pdf

ScyllaDB and spark:

https://www.scylladb.com/

https://docs.scylladb.com/kb/scylla-and-spark-integration/

TimescaleDB

TimescaleDB is an open-source database designed to make SQL scalable for time-series data

https://github.com/timescale/timescaledb

More relevant tools

Hive and Spark

Spark Hosting: https://towardsdatascience.com/diy-apache-spark-docker-bb4f11c10d24

DataFrame vs DataSet:
https://databricks.com/blog/2016/07/14/a-tale-of-three-apache-spark-apis-rdds-dataframes-and-datasets.html

https://tdwi.org/articles/2017/12/04/arch-all-data-time-and-the-data-lake.aspx

Data Serialization:
https://www.bizety.com/2019/04/02/data-serialization-protocol-buffers-vs-thrift-vs-avro/
https://stackoverflow.com/questions/40968303/thrift-avro-protocolbuffers-are-they-all-dead

Setting up Hive: https://www.edureka.co/blog/apache-hive-installation-on-ubuntu
Hive commands: https://www.dezyre.com/hadoop-tutorial/hive-commands

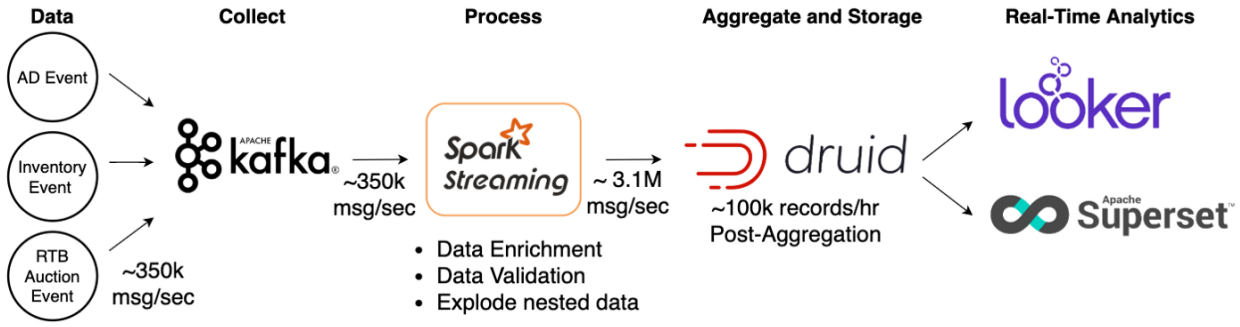
Apache Druid or Apache Kylin

Those are dedicated systems for Data Warehouses. Here is a good overview about these two and alternatives: https://en.wikipedia.org/wiki/Comparison_of_OLAP_servers

Blogpost about differences and pros/cons:
http://kylin.apache.org/blog/2018/12/12/why-did-meituan-develop-kylin-on-druid-part1-of-2/

Talk about how Airbnb uses Spark, Airflow and Druid:
https://www.datacouncil.ai/talks/how-superset-and-druid-power-real-time-analytics-at-airbnb

Use minion as deepstorage:
https://dzone.com/articles/how-to-configure-druid-to-use-minio-as-deep-storag



See
https://medium.com/gumgum-tech/optimized-real-time-analytics-using-spark-streaming-and-apache-druid-d872a86ed99d

Using cassandra for data lake:

http://cassandra.apache.org/ ?
https://de.slideshare.net/patrickmcfadin/analyzing-time-series-data-with-apache-spark-and-cassandra
https://thelastpickle.com/blog/2017/08/02/time-series-data-modeling-massive-scale.html

Advantages: Cassandra and spark work good in combination and there are a couple of examples.

Do we actually need a data lake? Maybe to store html documents from crawler, before it will be processed further?

Saving time series data with influxdb:

https://www.influxdata.com/blog/zeppelin-spark-and-influxdb-for-big-data-time-series-scenarios/
Data lake for storing raw data: https://github.com/influxdata/influxdb + telegraf for input
Data warehouse: Apache Spark

Influxdb for storing data in warehouse and using spark streaming as source:
https://medium.com/analytics-vidhya/processing-time-series-data-in-real-time-with-influxdb-and-structured-streaming-d1864154cf8b

https://medium.com/@xaviergeerinck/building-a-real-time-streaming-dashboard-with-spark-grafana-chronograf-and-influxdb-e262b68087de

https://datasciencechalktalk.com/2019/07/17/streaming-analysis-with-kafka-influxdb-and-grafana/

Cons: Influxdb comes in 2 versions as a solution, open source which can be installed only on a single instance and enterprise edition, which is paid and can be installed on a cluster.

Advantage: Easy integration for dashboards (grafana), telegraf for sources:
https://docs.influxdata.com/telegraf/v1.14/ (looks very promising in theory)

Spark Parquet as Data Warehouse

https://parquet.apache.org/

Flint

Using Flint, a time series library for apache spark:
https://databricks.com/blog/2018/09/11/introducing-flint-a-time-series-library-for-apache-spark.html
Downside: Not very active, and not working with spark 3.0:
https://github.com/twosigma/flint/issues/83

Data concept

Data sources, data structure, data tables

Variables and Success Metrics by URL and TIME

https://docs.google.com/spreadsheets/d/1UIwxvKds996FOvx372IN6WLteRD3rR3QGMCIxihc7NA/edit#gid=0

Data sources

- Google Search Console API
  https://github.com/topics/google-search-console
- Google Analytics API
  https://github.com/topics/google-analytics-api
- Matomo API
  https://developer.matomo.org/guides/reporting-api-clients
- Matomo Log Analytics https://matomo.org/log-analytics/
- Lighthouse API
  https://lighthouse.acolono.rocks
- Crawler
- HumanTxt API
- Securityheaders API

Data storage concept:
https://docs.google.com/document/d/1suwO7b41N29x7uIq2ZwNqtv1J3-v2WKFqHnm0nuZCr8/edit

Matomo Data

Userstory: As an acona end user I want to access my matomo data, process it and store it in the acona data warehouse, so that it can be used to calculate success metrics or evaluate rules to display recommendations.

We use Matomo API with Action Metrics like defined here:
https://docs.google.com/document/d/1suwO7b41N29x7uIq2ZwNqtv1J3-v2WKFqHnm0nuZCr8/edit#heading=h.phx19kyliuoe

Authentication: Hardcoded for now, but later it will be provided by the Acona Backend.

Variables: Matomo endpoint, matomo authentication key (later), interval (24h), urls, clientid

Result: Processed variables are stored in the data warehouse in our data structure.

Prototype:
https://medium.com/teamzerolabs/write-aprometheus-api-exporter-in-record-time-7518ee22f238

Prerequisites: Data Warehouse must be available and data schema must be defined.

ACONA Document analyzer

Goal: From a html document (e.g. provided by a crawler) specific metrics are processed (like number of H1, number of characters, …).

Input: HTML Document
Output: Metrics defined in https://docs.google.com/spreadsheets/d/1UIwxvKds996FOvx372IN6WLteRD3rR3QGMClxihc7NA/edit#gid=0

Used by: Scheduler uses Document analyzer as a job and then saves the result time-based for a specific url (See section for Jobs and Scheduler). The DAG can be started regularly.

Ressources:
https://tomassetti.me/parsing-html/
https://github.com/madeITBelgium/SEO-Analyzer
Term Frequency-Inverse Document Frequency or TF-IDF: "Building an ETL Pipeline in Python" von Daniel Foley
https://towardsdatascience.com/building-an-etl-pipeline-in-python-f96845089635

ACONA Recommendation Engine

The Acona recommendation engine provides rules that can be checked for a specific URL. It then returns recommendations, what should be improved.

There are two types of rules:
- Default Rules
- Specific Rules (coming from intelligence Suite)

Default Rules are defined here:
https://docs.google.com/spreadsheets/d/1UIwxvKds996FOvx372IN6WLteRD3rR3QGMClxihc7NA/edit?pli=1#gid=813777100

Storage: https://jsonlogic.com/

- In its own postgres database (https://www.cloudbees.com/blog/unleash-the-power-of-storing-json-in-postgres/) or

- in the Drupal Admin Backend for now (https://www.drupal.org/project/jsonb or https://www.drupal.org/project/json_field). That would have the benefit that we could add translatable strings like recommendations and use Drupal to maintain/update it; or
- as a pure json file in a github repo.

Tool to use defined rules to calculate results for a specific time and url:
Note, that there are already libraries in multiple programming languages to evaluate rules in jsonlogic, like `jsonLogic.apply(rules, data);`

The following input/output concept is based on an api approach, where results are processed somehow on the fly. Another approach would be to store results for every url, time(day) and variable (in the data warehouse) and another tool is accessing this data to return recommendations (the client itself or better its own “microservice” that is providing this data via API).

Input:
- URL
- Category: Optional, just use rules for a specific category (Html, Performance, …)
- Relevance: Optional, just use rules with a specific relevance (1-3)
- Language: Optional, if given recommendations for just a language are returned
- Variables: Optional, if given just rules for the given variables are checked. Format: `[{"html_h1_count", "html_h2_count"}]`
- Values: Optional, if given just rules with these variables will be checked with its values. In this case no database with latest values needs to be queried. Format: `[{"html_h1_count": "2", "html_h2_count": "4"}]`

Output:
All processed rules with its results, like:

```
{
  "timestamp": "1600872956"
  "rules": [{
    "id": "h1_count_red",
    "variable": "html_h1_count",
    "result": "green",
    "category": "content",
    "recommendation": {
      "en": "There should be exactly one H1 on a page. You currently have 2 H1 elements. You should change this.",
      "de": "Nur ein H1 Element soll pro Seite vorhanden sein. Stellen Sie sicher dass dies der Fall ist. Aktuelle Anzahl: 2."
    },
    …
```

```
  ]
}
```

Used by:

- Acona CMS Integration to display and save recommendations

Open Questions:

- Probably we will also need some authentication, so that this service can not be used by everybody.
- Where are rules stored? We can maintain rules in a Drupal database and make it available via API. Or should it be in the tool itself?
- When values are not given in the API request as argument, the latest values need to be queried from the central database.

xACONA Intelligence Suite

- Validate default rules (is rule XY really relevant for my project?)
- Use Data to calculate success factors (What page variables are relevant in my project? What page variables correlate with my page success?)
- create recommendations (“Try to increase metric XY, data shows that your most successful pages have at least a value of …”)
- Use Data for predictions (e.g. visits for a Spargel recipe will increase in April by 200%).
- Create warnings when there are anomalies (e.g. Visits for the Spargel recipe did not increase as expected, There is suddenly a decline in the number of Organic Clicks for a special page)
- make this notifications available via Push/pull

As a data scientist I want to access (historic) data for a specific domain, so that I can try to expose success factors or predict future metric values.

As a data scientist I want to have a working environment that helps me to perform my data analysis.

As a data scientist I want to know how I can use successful models to create new rules or recommendations.

How to set up Jupyter notebook for remote Servers:
https://www.digitalocean.com/community/tutorials/how-to-install-run-connect-to-jupyter-notebook-on-remote-server
Alternative: https://github.com/Bayesnote/Bayesnote

https://moz.com/learn/seo/seo-cheat-sheet
https://www.seo-suedwest.de/5825-google-auch-inhalte-mit-wenig-text-koennen-gute-rankings-erzielen.html

Flint:
https://databricks.com/blog/2018/09/11/introducing-flint-a-time-series-library-for-apache-spark.html

Resources:
Predictive Analytics:
https://medium.com/analytics-for-humans/a-comprehensive-guide-to-predictive-analytics-d1eb688f37dd
https://www.cardinalpath.com/blog/demystifying-predictive-analytics

ACONA Administration Tool

A central page where users can log in and update their settings.

As an ACONA client I want to log in on a central site where I can update my settings, authenticate a third party provider and choose my plan.

Requirements:

1. **User login, Permissions/Authentication** (only access/see own data)
   As an ACONA client I want to register and log in on a central site where I can update my settings.

2. **User handling for administrators**
   As an ACONA administrator I want to see what users have registered and maintain user status.

3. **Configure authentications** (third party access like to own matomo instance or google search console): Add key to matomo.
   As an ACONA client I want to authenticate third party services like matomo or google search console.

4. **Configure settings like domain/urls**
   As an ACONA client I want to define specific urls or a complete domain that should be tracked. For every url I want to assign a page type.

5. **Configure page types and the ACONA Success Score**
   As an ACONA client I want to specify my personal success metrics for different page types.

   Page Type 1 Name: Landing Page
   Page Type 1 Metric 1: Organic Clicks
   Page Type 1 Metric 1 Type: Number (max 10.000), Number (max 1.000), Number (max 100), Number (max 10), Number (max 1), Percent, Boolean
   Page Type 1 Metric 1 Relevance: 1 - 100 (from last to most relevant): 90
   Page Type 1 Metric 2: Conversion Rate
   Page Type 1 Metric 2 Type: Percent
   Page Type 1 Metric 2 Relevance: 1 - 100 (from last to most relevant): 50

   Example: For a given datetime the Number of Organic Clicks is 400 (Max 1.000) and the Conversion Rate is 0.25 (25%). So the ACONA Success Score is: 400 / 1000 * 90 + 0.25 * 50 = 48.5

6. Configure plan (later, Basic Plan for now)
   As an ACONA client I want to be able to choose a payment plan.
7. Maybe payments (later)
8. Provide user data and settings via API

9. **Storing rules and making them available via API**. See recommendation engine. Rules storage can be also done using an external json storage. https://jsonlogic.com/
10. Provide oauth server, so user can login in external systems like grafana with drupal login.

Content architecture:

- User: name, e-mail
- Domain/Group: name, user/author, variables, services, urls
- URL: name/link, pagetype, user/author
- Variable: name, service
- Service: name, url, key
- Key
- Rule: Title, Recommendation, Category, “Relevance”, “More Info”

- Rule evaluation: Rule, Logic, Indication
- Page Type: e.g. “Infopage”, “Landing page”, “Commerce Product”
  Name, Page Type Number, Author/user, Success Score definition (Variable 1, Relevance 1, Variable Type1, Variable 2, Relevance 2, Variable Type2, Variable 3, Relevance 3, Variable Type3)
  Note: Extra validation needed: Only one page type by page type number and user possible; Extra warning message when changing data (history data will be not comparable any more)

Vocabularies:

- Category: e.g. “Performance”, “UX”
- Page Type ID: “Type1”, “Type2”, “Type3”
- Service Types
- Units

API endpoints:

- Active Users
- Page types with Success Score by user/domain
- Rule evaluations with Metadata (Rules)
- URLS with page type by user/domain
- Active Variables by user/domain

Open questions:

- When calculation of the acona success metric changes, what about the historic data? Will it just stay as it is? Yes
- Include dashboard functionality there? (probably not for the start)

Technical solution:

- Using Drupal for the login and for storing all relevant settings.
- Dashboard solution: Cube.js (cube.dev)? Grafana? Could be an extra instance with sso from Drupal.
- Matomo goals can be queried via API: https://developer.matomo.org/api-reference/reporting-api#Goals Those goals can then be listed as success metrics.

ACONA CMS Integration

- Setting page where a acona id/key has to be given

- Overview: List of URLs with current Acona Success Score and number of recommendations
- **By url: Display most current success score (=MVP)** and a history of one week/month/year
- By url: Display recommendations (rules validations)
- Display Page specific stats and notifications on Page Edit screens
- Later: add a Button to analyze specific Pages immediately (maybe also during writing)

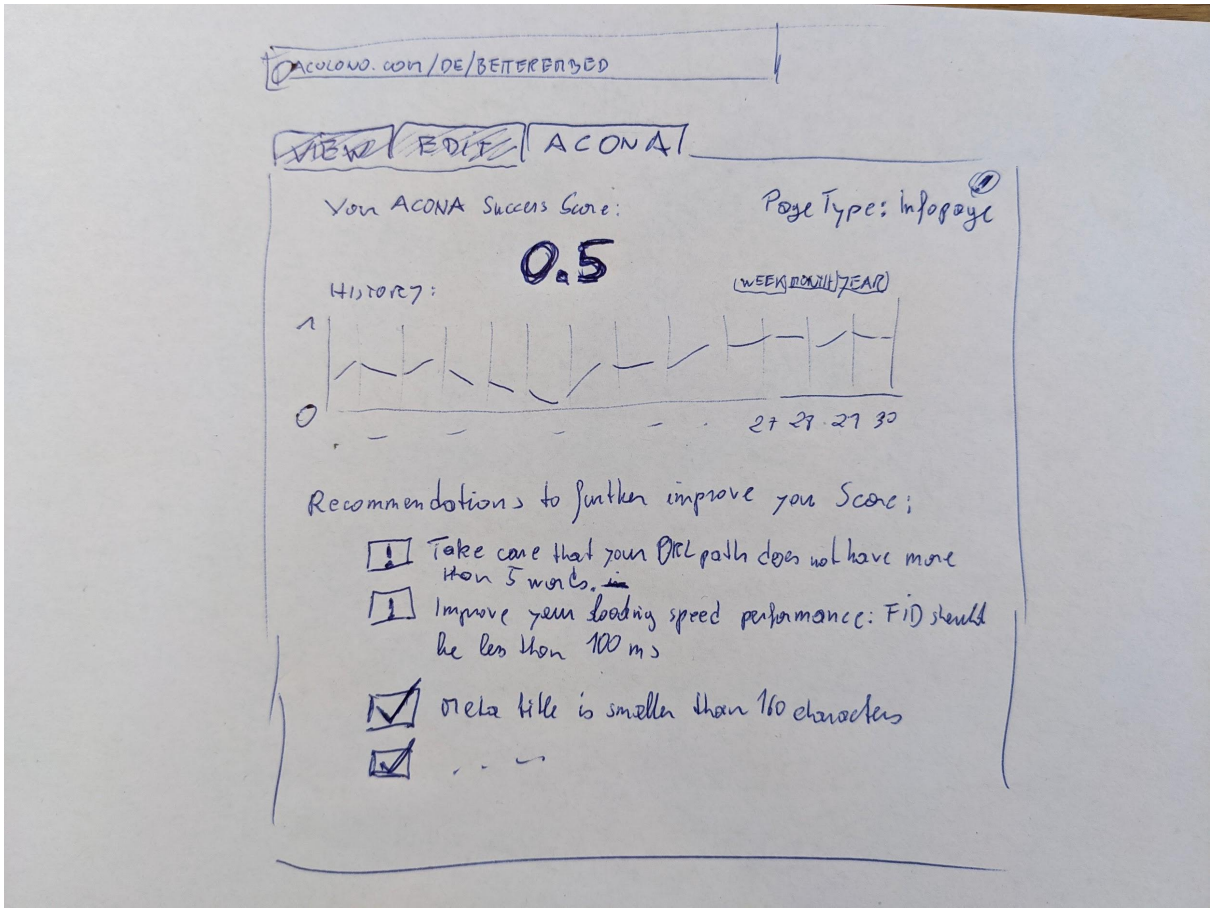
The cms integration would need the following data:

- **Get urls and their page type for a user id and domain (stored in CMS module settings)**
- **Get current (last) ACONA success value by url and page type**
- **Get history for ACONA success value by url, page type and date (day)**
- Get names for the 1-4 defined page types
- **Get all rules that are evaluated true by url and date (day)**
- **Get all rule metadata (title, recommendation, variable, relevance, indication, category, more info) by rule id.**
- **Get history for a specific variable by url and date (day)**
- **Get all active variables for a user id and domain.**

ACONA overview page:
Lists all active URLS and the current most important value (ACONA success score). Later also how many rules are evaluated true with indication green, red and yellow.

URL (Linked to detail page) - PAGE TYPE - SUCCESS SCORE - NUMBER RED RULES - NUMBER YELLOW RULES - NUMBER GREEN RULES - UPDATE TIME - UPDATED BY

ACONA URL specific detail page:



Later: Setting by page:

- Page type (One of 3-4 defined page types)
- Keywordphrase/Intention: Can be used to compare to google search console data and if it fits to title, body text and metadescription. Those metrics would have to be calculated on the fly then. (possible future feature)

Settings by client in CMS:

- Acona User ID (authentication)