*Last updated*  12.12.2019 • *by*

MindMyPen

Content

[1 Introduction 2](#_Toc22721442)

[1.1 Market comparison 2](#_Toc22721443)

[2 User stories 3](#_Toc22721444)

[2.1 Concepts 3](#_Toc22721445)

[3 Solutions 5](#_Toc22721446)

[3.1 Identified domains 5](#_Toc22721447)

[3.2 Identified entities 5](#_Toc22721448)

[3.3 Data structure 6](#_Toc22721449)

[3.4 Data examples 7](#_Toc22721450)

[3.5 Functionality 9](#_Toc22721451)

[3.6 Visual functionality 10](#_Toc22721452)

[3.7 Ideas for the future functionality 10](#_Toc22721453)

[3.8 Architecture 11](#_Toc22721454)

[4 Product description 14](#_Toc22721455)

[4.1 Entities 14](#_Toc22721456)

[4.2 Storage 16](#_Toc22721457)

[5 Detailed specification 17](#_Toc22721458)

[5.1 Content and functional requirements **Fejl! Bogmærke er ikke defineret.**](#_Toc22721459)

# Introduction

A tool for authors with an interest in complex storylines. Also, a tool for analyzing other titles and map how different aspects are covered throughout the work.

In comparison with other tools, this tool focuses on the complex relation between the ideas and its textual representation as to offer a visual representation on how a complex matter is delivered in text.

## Market comparison

|  |  |  |
| --- | --- | --- |
|  |  |  |
| http://dramatica.com/design/images/dramatica-logo.jpg | [Dramatica](http://dramatica.com/purchase) | Desktop. Focuses on supporting the user in developing a story. As opposed to MMP which only helps support the creative process |
| https://www.storyist.com/assets/images/Storyist-Icon-48.png | [Storyist](http://storyist.com/) | Desktop. Ios only. Good PLOT OVERVIEW. |
| Story Planner | [Story Planner](http://www.storyplannerapp.com/) | Desktop Ios only. Too simple. |
| [a cat] | [Save the Cat](https://store.savethecat.com/collections/products) | Desktop and app. Focus on screenwriting. Ugly! A lot of focus on "method". |
| http://truby.com/wp-content/uploads/2012/12/trubylogo.png | [Truby’s tools](http://truby.com/blockbuster-2/) | Desktop. Ugly! A lot of focus on "method". |
| C:\Users\md\AppData\Local\Microsoft\Windows\INetCache\Content.MSO\BBBD3A7A.tmp | [Novel Software](https://www.novel-software.com/) | Web and desktop. Has "Advanced Plot Point Tracking" |
| C:\Users\md\AppData\Local\Microsoft\Windows\INetCache\Content.MSO\3D18E88C.tmp | [Scrivener & Scrapple](https://www.literatureandlatte.com/) | Desktop. Keeps track of details. |
| C:\Users\md\AppData\Local\Microsoft\Windows\INetCache\Content.MSO\B1B8D898.tmp | [Evernote](https://evernote.com/intl/da/) | Collections. Online. API!!! |
| C:\Users\md\AppData\Local\Microsoft\Windows\INetCache\Content.MSO\6AA094F0.tmp | OneNote | Collections . Much like Evernote, but free for most people and part of the MS ecosystem. Good for alternative off/onloading. |
| https://www.finaldraft.com/wp-content/themes/Divi-Space-Child/images/fd11_logo.png | [Final draft](https://store.finaldraft.com/all-final-draft-products.html) | Desktop. Screenwriting focus (See REFINED BEAT BOARD). |

# User stories

## Activities

### Searching for related content

1. Build a search string
2. Maybe select among searches performed earlier
3. Execute it
4. See all findings in a collapsed tree of sections with count on each section and each level
5. See total count and what we have been looking for

## Concepts

### Works and sections

All text contents are a combination of sections and it should be possible to create works from sections and maybe allow sections to be included in multiple works. When changing a section that is used in multiple works, the user should be informed, giving a possibility to version a section or split it. Later maybe it should be allowed to merge a split section again.

Also, it should be allowed to tag and group sections so to easier organize them as well as an extensive support for searching and visualize search results.

### Aspect

Aspects that are central for the story and are spread out over the storyline but doesn't make up the actual storyline. Different aspects must be cared for in different ways and can therefore be related to in multiple different ways and it can be necessary to track that the aspect is treated in an interesting way and with regards to the dramaturgy of the story. Aspects can be tagged in the same way as sections

### Story levels

Story levels are multiple layers of aspects that are both independent and interlinked. It is extremely important that a level is consistent in the way it serves the aspects of the story. Is it backstory, is it key plot elements, is it action peaks, is it the day-to-day activities etc. There are relations between the aspects from the different levels, and it should be possible to visualize this.

### Aspect analysis

In an existing text, the Aspects are not known up-front, but must be found. Suck aspects should be found automatically and edited manually.

### Details

Apart form aspects there are many details that are much more practical, like people, places, time and grouping of this. These elements are all interrelated. It is often the same things, but to some extend it looks like aspects, as we can group and relate the aspects and contain both a descriptive structure linked to actual instances or examples in the text.

Some structures are very common like, people, places etc. This should be templates or special aspects or something separate.

### Dramaturgy (Sections and acts)

All text has different sections. Simple reports have an intro, a body and a conclusion, while a novel can have many different sections. The sections are extremely important for the Dramaturgy or the overall quality of the text. Therefore, visualization of how the various aspects are treated in the different sections are a vital tool as well as the possibility to being able to thoroughly distinguish the different sections from each other. Should support different views and colours for both sections and aspects and details.

### Share content

It should be possible to share the content with other persons and even work on the same material together. Not the same section though. You could need various working groups or compile a version and give persons access to a specific version.

### Versioning

It should be possible to version all sections and an entire work.

### Relating details and images

Especially for characters and places, it would be nice to be able to link images and maybe descriptions and links to online content. Also, a colour code and a mini-image used in visualization would be good to identify characters from each other.

### Analysis

It would be interesting to be able to analyse the text for e.g. Lix number page and word count, spelling, aspects, relations etc.

Of cause page count would be based on a specific page size etc.

### Scenes

The story is not told laniary or is structed by the details of the reality it describes. It is described by scenes. The scenes create the energy and the narrative quality of the story. So, the story is built from scenes while every scene is packed with details from persons, places and all kinds of aspects. The scene's function is to describe the relations of all the aspects and open and close questions about where the story is moving. Therefore, the scenes are the working area of the story building and should be like a controller or a desktop from where there is easy access and overview of all the aspects, and what is included, how it is dealt with, what is changed. It should be possible to see a clear pre and post condition of the entire story as well as a sort of input/output schema. All scenes eat some of the story (puts it in its place) and creates new details – questions.

Scenes can be divided into beats (A phrase from screenwriting). It should be optional whether such detailed level should be used. It might get too technical when it relates to novels and not movies

### Visualization

It is important to understand that the use of "Aspects" is an effort to make the tool so generic as possible when it comes to the individual writer's practices. Yet the core quality of aspects is the possibility to maintain relations throughout the material to get some sort of overview across the common building stones like acts, sections etc. We as constructors of the tool don't know what the user is looking for, but we know that it can be essential or at least an interesting tool to create some insight into the final work or work in progress. So visualization of relations is important both to understand the quality of the tool as well as enhancing the interest for using the tool. Finally, an expert user of cause expects the use of aspects to help where it most important, so display a graph of the complexity derived from the nitty gritty text so it is comparable only to the work from a structural perspective, It s size, sections, acts and chapters. Therefore, visualization is like a trademark of the entire product/tool and must have much emphasis.

### Templates

For both dramas, novels, short stories and even a thesis, there is a common structure. Therefor it is obvious to start out with a template including acts and/or phases and story levels. Depending upon how the aspects are being used, it makes good sense to allow for a writer to start out with a template which can be modified for the specific needs of the work.

### Mobility

Writers don't write on their phone, but they would like to see structures and they would like to narrate by voice and categorize their recordings, so it would be easy to identify it later. Also, some sort of speech to text (SoT) translation is an obvious function. SoT might not be used extensively, but as mock-ups to be cleaned up later or used as a template makes sense, at least it is a functionality that most would try out and then it is a personal choice to what extend and how this is utilized.

### Diversity of use

No writer works in the exact same way and therefore, the process should not be made hard by introducing too many aspects before getting to the main task which is writing. Writing is the first thing a person can do. Later that entry or those entries can be moved into a work. The work can be a simple collection of entries. Only later the user might want to introduce acts, chapters and beats, or maybe only chapters.

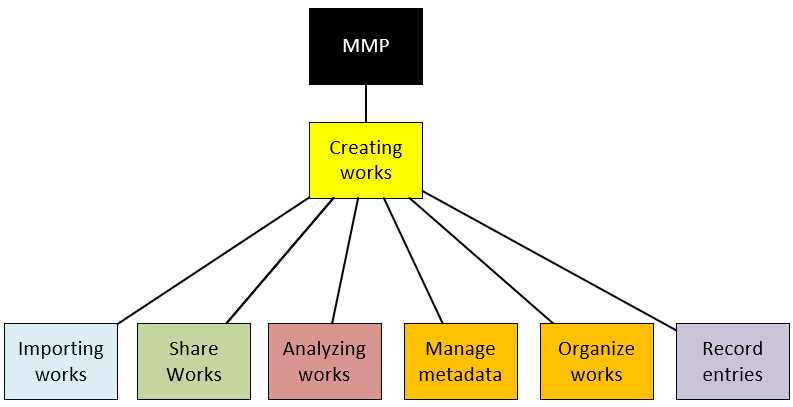
Aspects, persons etc. might only be relevant for the writer to organize later, and it would be cool to do some sort of reverse development, where analysis can be used to dig out all the aspects that you want to organize incl. persons.

### Technical papers and studying

A very interesting way to use this advanced way of working is to support education and research. This should not be an initial focus, but it would be nice to be able to import notes from e.g. www.diigo.com.

# Solutions

## Identified domains



## Identified entities

|  |  |  |
| --- | --- | --- |
| Type | Entity | Description |
| O | Work |  |
| O | Chapter | … |
| O | Scene |  |
| O | Beat | A section of a scene that is always characterized by a pre and postcondition. Very much like scenes and acts |
| O | Character |  |
| O | Workgroup |  |
| O | Release |  |
| O | Location |  |
| O | Recording |  |
| O | Artifact | Images and links relating to persons, places and aspects |
| O | Analysis | A recording of an analysis |
| O | Entry | Snippet of text |
| O/D | Tag |  |
| S | Version |  |
| S | Fork | A text |
| S | State | Capture the state of all elements relating to an entire work, including the text, aspects, characters and places |
| D | Act | … |
| D | Phase | Dramaturgy building blocks |
| D | Aspect |  |
| D | Aspect stage |  |
| D | Aspect relation |  |
| D | Aspect group |  |
| D | Story level |  |
| O | Story event | Something that happens in a story level |

O: operation data

D: Mainly descriptive data compared to the operative data

S: State data relating to the operational data

## Data structure

### Aspect relations

All the story events are actually aspects that you dive into while telling the story. While they are referred to as events, there are many other details that are interlinked. Like information about the persons, their location and other events. Therefore, it should be possible to link any part of the story (From act to beat) to any aspect and identify that link with a name or a stage of some sort. The same should be possible between aspects.

The stages or relation types that comes out of this is created in the same way you create tags – by grouping similar expressions and offer them as an option once they have been used the first time.

As an alternative You could restrict a set of values to include only a preset number of options.

### Structuring

Structure is second to content. Therefore, any entries can exist without any additional structure. Yet it should be possible to combine any structure and relate the entries thereto.

* Series
* Work
* Short story
* Act
* Phase
* Chapter
* Date and time
* Scene
* Beat

The listed levels can be turned on and off and entries can be added at any level. When a level is added, all entries are of cause moved to the first instance of a new level under every parent level (If any). If a level is removed, their child elements will of cause be grouped under the parent.

It should also be possible to just hide the individual elements. By zooming in and out in a way.

If an entry should be divided into two different beats, it should be possible to do so simply by breaking it up between two paragraphs.

At the lowest level, it is not possible to start or end a section inside two different entries. The section can span multiple sections from start to end, but not to specific positions inside the entries. Within the same entry, it is different. Here it is possible to add multiple sections to various sections. So, in short:

* An entry can be a section.
* A section can contain a series of entries
* An entry can contain multiple sections
* The last section has no length – it goes to the end of the entry or end of parent section

### How to identify sections across entries

To avoid splitting entries up because it is being sectionalized, the sections positions are defined by the character position in an entry. When an entry is changed, the position is updated. Positions are only saved for the lowest level sections. Sections at a higher level will always depend on lower laying sections. In series of sections (Siblings) only the first will have a start position (Usually 0) unless a parent section exists, in which it will borrow its position. In that way we have as few positions to update as possible. While only the first sibling section has a start position, all sections will have a length. When an entry is changed, we find the single section to be modified, and usually we only need to change its size, for all other sections to match. This of cause means that the text can't be edited outside, unless it is exported with codes defining the sections.

### Versioning

Every change to an entry is logged as a delta using e.g. git repo google/diff-match-patch while keeping the latest result complete, while the previous versions only contains the reverse changes (if possible).

MMP should make a backup every e.g. 5th minute or when a user finishes work.

If an entry is changed, any section positions affected by the changes must also be saved with the version. That is typically only start positions and lengths of sections that start/stop within the area that changes has happened to.

You shouldn't need the possibility to explicitly save a version of an individual entry, yet when working with a Work, you should be allowed to create a version of the entire work including sections and all other details (Also aspects etc.). Some of these versions can eventually become "Releases".

### Entries

The core element of MMP is the entries. Entries are not organized in any works from start. They are simply a collection, much as entries in a diary or journal. When creating a work, You would work with a few big chunks of entries. Obviously, if you apply structure, moving the entries around will create chaos. Once structure is applied You would more likely move text around defined by the structure, rather than the start and end of the individual entries. The individuality of entries should disappear when it is accessed trough a work. In the work, it is the sections that you use to define individuality. Therefore, when an entry is added to a work, one large section is created immediately to allow to sub sections to be created.

There is a difference to the way entries are used depending upon the type of work being created. If it is a journal, you would "click" [Add new entry], while when working with a Novel. You would "click" [Add new Chapter] or [Beat].

### Aspects relating to entries

Aspects are linked to the entries in one of two ways. Either by a phrase that are the same as the aspect title (e.g. a person) or similar. This could be specified by using a combined regular expression like (expression1| expression2|expression3).

More specifically would be to specify an exact position by a start and length like a section. This would create a special section at the lowest level called "Aspect section". Alternatively, Aspects should be linked to existing sections like Chapters or Bets.

When specifying the relation by a regular expression, it would be good to create the instances as sections as well, but only for caching as it can take up unnecessary space.

### Accounts and sharing

All users have an account and all entries and works are bound to an account. An account is owned by one user. It is possible to share a version of a work with other users. The users are identified by OAuth only. No support for password-based login.

### Voice recording

Recordings should be an integral part of a work. You should be able to store multiple recordings and name and describe them. Recordings should be an integral part of creating a work in conjunction with text. Maybe it should be a type of entry. For now, though, voice recording will be handled separately. A voice recording is a separate entity. When transforming it to text, the text is linked to the recording as an entry, so entries in general should be linkable to a soundfile. A recording can be converted multiple times using improved technology over time. Every new recording creates a new version.

## Data examples

* Work
  + Aspect: Mysteries (Group, not ordered)
    - Mysterie 1
      * Phases:
        + Første benævnelse
        + Personlig reaktion
        + Forklaring
  + Aspect: Dramaturgy
    - Præmis
    - Spændings definition
    - Styrende idé
    - Pitch
    - Red.pith
    - Værdier
    - Polaritetsrejse
    - Temaer
    - Virkemidler
    - Roller overordnet
    - tid+ ramme overordnet
    - Genre
    - Fortælleform
    - Format
  + Aspect: Characters (Group, not ordered)
    - Main character
    - Villain
  + Aspect: Locations (Group, not ordered)
    - Location 1
    - Location 2
* Story levels
  + Baggrund
  + Situation
  + Begivenheder
  + Rejse
  + Vendepunkter
* Section levels
  + Story
  + Act
  + Phase
  + Chapter
  + Scene
  + Bede
* Sections
  + Act 1
    - Indledning
    - Exposition
    - Point of Attack
    - Kollision
    - Calling from God
  + Act 2
    - Symbolsk scene
    - Midter plot punkt
    - Point of no return
    - Fortrydelsespunkt
    - Stop fortællingen
    - Plot punkt
  + Act 3
    - Vendepunkt i 3. akt
    - Falsk slutning
    - Sand slutning

## Functionality

|  |  |
| --- | --- |
| Domain | Functionality |
| Creating Work | * Create new Work (Auto add first entry) + Remove * Copy details from template * Save work as template * Create new Journal (Special Work) * Add new entry * Copy entry to another Work (And create link) * Move Entry to another Work |
| Sections | * Add new Section (And entry in it) (Type based on current level * Apply Section (Over existing entries) + Remove * Move sections up/down * Combine/Split sections * Split entries + Merge |
| Aspects | * Create Aspect + Change + Remove + Reorder (Under work or parent) * Add Aspect Property + Change + Remove + Reorder * Add Aspect Phase + Change + Remove + Reorder * Link Aspects + Remove * Add Aspect to Entry (Maybe set Phase) + Remove |
| Story levels | * Add story level * Enable disable story level * Change order of story levels * Add Aspect to story level |
| Importing | * Join with Diigo account + Change + Remove * Import to Work (All tags + No tags) + Add Entry and/or add link * Upload RTF file and create Work and entries * Upload Word file and create Work and entries |
| Exporting | * Export PDF * Export EPUB * Export Word * Export RTF |
| Share | * Create and share state with otherers * Invite user (While publishing state) * Share with existing user + Remove * Share new release with those already shared with |
| Account | * Sign up (OAuth + personal details) + Change details * Sign in * Close account |
| Analyze | * Analyze work * Analyze every entry * Analyze every section at some level * Analyze   + HDInsight (MS):     - Word count     - Character count   + <https://github.com/MarkBuskbjerg/tekstr>     - Lix calculation       * MARKER # LANGE ORD       * MARKER # LANGE SÆTNINGER       * MARKER # MEGET LANGE SÆTNINGER   + Homemade:     - Compare size with known publication sizes     - Calculate nof. pages of known publishing formats   + Text Analytics API (MS):     - Get locations, names and topics   + Azure Bing (MS)     - Search     - Auto correct spelling   + Azure Translator Text (MS)     - Auto translate (Save as new version identify by language) |
| Record | * Record (And save temp + Save) + Change details + Remove * Upload recording to save * Playback * Convert recording to entry (And join) + Save (Maybe new version) * Add recording to entry + Remove |

## Visual functionality

|  |  |
| --- | --- |
| Domain | Visual functionalit |
| Creating work | * Find all instances of a phrase (Support regex) |
| Sections | * Scroll down/up * Collapse/Expand single/All sections |
| Aspects |  |
| Story levels |  |
| Importing |  |
| Exporting |  |
| Share |  |
| Account |  |
| Analyze |  |
| Record |  |

## Ideas for the future functionality

There are many ideas for functionality we want to implement, but it is important that it is given little emphasis, until time is mature.

1. Payment
2. Allow and track multiple people working on the same text
3. Allow for images a.o. resources inside the entries
4. Commenting on works by 3rd party users
5. Maybe use Tags
6. Usersettings for used fonts and colors etc.
7. Maybe use recordings exactly like entries
8. Mindmaps of aspects
9. Multiple mediatypes in entries
10. Support working with technical papers, both short and long.
11. Support importing and analysing documents
12. Store exported versions
13. Support Diigo import
14. Suppor forking and merging entries
15. Export EPUB version
16. Analyse text and autocreate aspects and sections
17. Identify spelling errors and offer to fix them displaying a list
18. Support progress states on individual sections or entries
19. Support different formatting of dialog, reading a letter, custom 1, 2 etc.
20. Each user should have teir own storage account
21. Allow for a local installation and/or local datastore with data gateway
22. Save exported files in a fileshare
23. Add tags to entries and support searching it

### Architecture

1. Join tables Validate JSON responses towards JSON Schema (https://www.newtonsoft.com/jsonschema/help/html/GeneratingSchemas.htm)
2. Autogenerate schemas based on classes after new version release. Maybe contain multiple versions

## Architecture

Online solution with minimal mobile support hosted by Azure:

### Frontend

* Single Page Application based on Angular (Angular is selected over React, because we want a whole framework)
* [Kendo](https://www.telerik.com/kendo-angular-ui) for Angular visual component
* Backend content is a static web page in a storage account while all functionality is functions or logic apps. Access is managed through an API management service.
* If required CDN is used to ensure minimal response time concerning static content and selected searches based on the URL and arguments.

### Handling data in client and in backend

En Service fungerer som store. Den besnytter en service der opdaterer server: Synchronizer Andre services

**Synchronizer**:

Sends changes on to server at a steady rate

AMAZON POLLY!!!!!

All changes are send to store in a package that stagtes:

### Security

* Users are authenticated by OAuth only and added to the AD.
* All content is encrypted at rest and only readable on the owner's desktop. This is solved with the use of Storage Service Encryption, which uses 256-bit AES encryption that aught to be more than enough.
* Personal keys are used so only the owner can read text values. We will use **Customer-provided keys** as opposed to Customer-managed keys that doesn't ensure fully exclusive access to own data. The key could be created from the user's email, salt and a keyword (As there is no password for OAuth). The Keyword could be handled using "Zero-knowledge password proof" and the SRP protocol to ensure, that what the only code that sees the plain text password is executed on the user's choice of browser/client.
* En-/Decryption can happen on the owner's client by using the JavaScript sdk azure-sdk-for-js and particularly key vault-keys for en-/decryption on the client to make sure that only the user can read his/her own data (<https://azure.github.io/azure-sdk-for-js/keyvault-keys/>)
* Only when needed for e.g. analysis, decryption can happen on the server. It requires that the encryption key (Not the Keyword) to the server.
* We can't avoid that security is breached if the code is changed to allow for data to be logged when decrypted on server, but the data can't be hacked or copied. In the end the end-user still needs to trust the solution provider.
* All web communication is of cause HTTPS.
* When data is passed to the client it is always encrypted and only decrypted on the client. When sending data to the server the data will be encrypted as well. Yet only the content while mark-up required to identify sections used to merge/persist data isn't. HTTPS can do this for us as well, so we can be laxer on this if forced to.

### Networking

* VNET with subnets are created
* Access via FTP and ExpressRoute

### Computing

* No computing in API layer. All should happen asynchronies
* Use functions and logic apps as much as possible
* Create asynchronies solutions through service bus

### Scaling

No need for any type of Application Gateway on the frontend.

We can enhance responsiveness for API management by

* Enabling Redis caching
* Expand nof. API mng. Instances (Change SKU)

Function responsiveness can be enhanced with

* Scaling based on nof. Calls
* Always on (Requires Premium plan)

Access to storage can be enhanced with

* CDN
* API management Redis
* Geo-redundant storage (GRS): Cross-regional replication for Azure Storage

### Storage

* Static website is stored as blob
* Recordings and entries are saved as blobs as well
* Accounts and Users are saved in tables
* Work and related descriptive entities are saved together in a table for each work as json
* Aspects and related entities are saved in another table as json by work and aspect
* Sections and related entities are saved in another table as json by work and section
* Entries (Without content) are saved in another table as json by work and entry
* Entry versions (Deltas) are saved in in another table by work and entry+time
* Changes to work, aspect and sections are send in blocks every 5th minute or so (Or on demand) and will be merged with the tables
* Changes to entries are send with the same interval:
  + Saves reverse changes for old version (fully encrypted)
  + Sends the new version to completely override old entry
  + Entries are by default clipped by chapters, so each chapter is a block blob
* Releases of an entire work involves a complete copy of all related entities incl. the entries – not as deltas, but as exact copies of the latest version.

## Technical considerations

### Online editing

#### Limited memory

We should limit text content to max. 2GB in memory at a time. The Lord of the rings trilogy with over 500K words has a size of approx. 5GB as RTF format. If text is being edited or otherwise processed, we should expect it to be duplicated in memory or even tripled somehow. SO we will need to work with smaller sections of a novel. E.g. a chapter at a time.

Some entries can span multiple chapters, because we won't limit a user to smaller entries, if they don't want to (Max. 1TB though). In that case we might have to download entries partially, which will be tricky. Yet we will solve this in the end, because it is just a question of math.

#### Edit Smaller portions

An RTF editor with formatting will start to be slow after 100K words. I have no knowledge of any chapter being that long, so an editor can be limited to work with one chapter at a time, or entries, if they are smaller.

When not all text is in the editor, obviously, the editors functionality apart form editing can't really be used. This is not a problem, because it is mainly word count and spell checking that needs to look trough the entire document, and that has nothing to do with editing, and this is something that JavaScript, an external service or the backend can do just as fast or faster.

#### Editors

**Inline editing**: CKEditor can change from read-only text to editor seeming less which is very cool. It also has an extensive API and is among the fastest components to load large content.

<https://ckeditor.com/docs/ckeditor5/latest/index.html>

#### Make limits configurable

Opera has just launched a gaming browser that can allow to extend (or limit) resource usage. In the future the memory management might allow for more configuration in general, so this limit should also be editable or set to auto-config.

#### Consider desktop solution

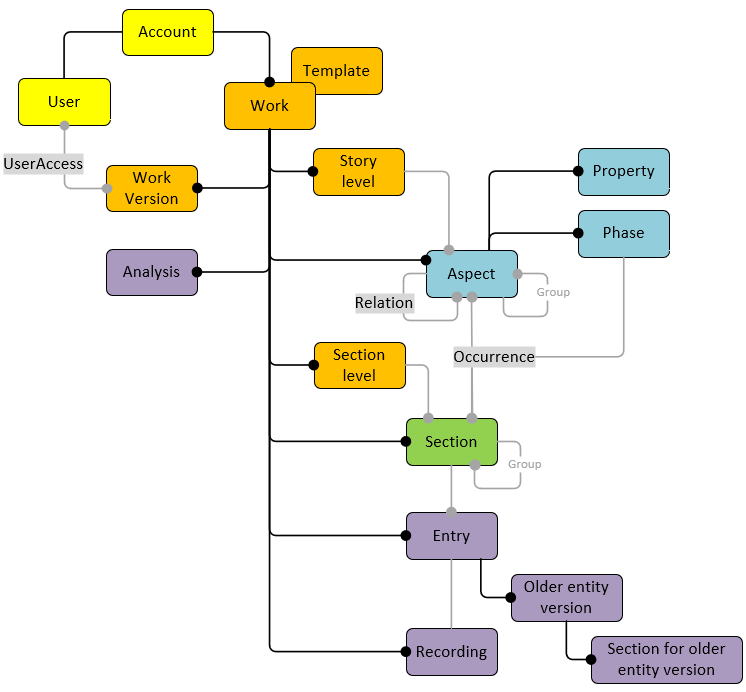
Obviouesly, the data size requirements makes it easier to offer the solution as a native desktop application. This might be possible combining Angular with electron. See more:

<https://developer.okta.com/blog/2019/03/20/build-desktop-app-with-angular-electron>

<https://github.com/AngularFirebase/angular-electron>

# Product description

## Entities

**

### Properties

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| |  |  | | --- | --- | | Entity name | Properties | | Section level | Title  Description  Color  IsDisabled  BreakEntry  Level | | Section | StartPosition  IsNewEntry  Entry🡪  Length | | AspectOccurrence | 🡨 Section  Aspect 🡪  Phase 🡪 | | Aspect | Title  Description  Color  Symbol  IsDisabled  IsSearchable  SearchString (Alternative to Name)  IsGroup  IsChildrenOrdered  OrderNo | | AspectRelation | 🡨 Aspect  Aspect 🡪  Title  Description | | AspectPhase | Title  Description | | AspectProperty | title  IsEnabled  OrderNo | | AspectPropertyValue | Value | | AspectPropertyArtefact | Link  Type | | Work | Title  Description  Version  Language  Type: Work, Journal, Template | | WorkRelease | Work 🡪  Date  Title | | Account | User 🡪  Level  State | | |  |  | | --- | --- | | Entity name | Properties | | User | ((OAuth only)) | | UserAccess | WorkRelease + Account 🡪  🡨User  AccessLevel | | Analysis | Entry  AnalysisType  Result | | Entry | Date  🡨Author  Subject  Extract  Last Changed  LastChangedBy  OrderNo  CopiedFromEntry + Work 🡪  Tags  Recording 🡪  Hyperlink (From e.g. Diigo) | | OlderEntryVersion | Diff  Date | | SectionPositionForOlderEntryVersion | 🡨Section  Entry🡪  Length  Position | | List of colors |  | | List of symbols (Icons) |  | |

### Settings

|  |  |
| --- | --- |
|  |  |
| *Account level* | People only login using OAuth from Facebook, google or LinkedIn. They are not forced to create a specific account, but they are pushed to extend their use of the product trough different service levels:   * **Limited**:Limited amount of content * **Secure**: Fully encrypted * **Pro**: Multi-use, import, Aspect properties * **Advanced**: AI support * **Integrated**: Support for local data * **Test**: … |
| *Account state* | * **Ready**:… * **Locked**: … * **Archived**: … * **Test**: … |
| *Work type* | * **Template** * **Work** * **Journal** |

## Storage

### Tables

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Table | Partition by | Row by | Data structure | Example (Expand to see) | |
| Account | (Global) | Account | Account  ⮡ User | {  "id": 1,  "accountlevel": "basic",  "state": "ready",  "user": {  "id":"oauth-id"  }  } |  |
| Work | Account | Work | Work …  ⮡ Story level  ⮡ Aspect  ⮡ Aspect relation  ⮡ Aspect (Children)  ⮡ Property  ⮡ Phase  ⮡ Work version  ⮡ User Access  ⮡ Section level  ⮡ Recording | {  "Title": "Novel template",  "Type": "work",  "Language": "Danish",  "Sectionlevels": [  {  "Id": 1,  "Title": "Work",  "Description": "...",  "Color": "AA1111",  "Level": 1  },  {  "Id": 2,  "Title": "Phase",  "Description": "...",  "Color": "11AA11",  "Level": 2  },  {  "Id": 3,  "Title": "Act",  "Description": "...",  "Color": "1111AA",  "Breakentry": 1,  "Level": 3  },  {  "Id": 4,  "Title": "Chapter",  "Description": "...",  "Color": "AAAA11",  "Breakentry": 1,  "Level": 4  },  {  "Id": 6,  "Title": "Scene",  "Description": "...",  "color": "66FF11",  "Level": 5  },  {  "Id": 5,  "Title": "Beat",  "Description": "...",  "Color": "AA11AA",  "Level": 6  }  ],  "Aspects": [  {  "Id": 1,  "Title": "Personer",  "Description": "...",  "Value": "",  "Color": "5511FF",  "Symbol": "...",  "IsEnabled": 1,  "IsSearchable": 1,  "Searchstring": "",  "IsGroup": 1,  "IsChildrenOrdered": 0,  "OrderNo": 1,  "Aspects": [  {  "Id": 2,  "Title": "Hovedperson: Anna",  "Description": "...",  "Value": "Anna",  "Color": "5511FF",  "Searchstring": "",  "Relations": [  {  "FromId": 5,  "ToId": 5,  "Title": "Nordvest",  "Description": "..."  }  ]  },  {  "Id": 3,  "Title": "Hovedpersons mand: Alfred",  "Description": "...",  "Value": "Alfred",  "Color": "55FFFF",  "Searchstring": ""  }  ]  },  {  "Id": 4,  "Title": "Steder",  "Description": "...",  ]  } | "Value": "",  "Color": "552244",  "Symbol": "...",  "IsEnabled": 1,  "IsSearchable": 1,  "Searchstring": "",  "IsGroup": 1,  "IsChildrenOrdered": 0,  "OrderNo": 2,  "Aspects": [  {  "Id": 5,  "Title": "Nordvest",  "Description": "...",  "Value": "Anna",  "Color": "5511FF",  "Searchstring": ""  },  {  "Id": 6,  "Title": "Annas arbejdsplads",  "Description": "...",  "Value": "Alfred",  "Color": "55FFFF",  "Searchstring": ""  }  ]  },  {  "Id": 7,  "Title": "Hemmeligheder",  "Description": "...",  "Value": "",  "Color": "FF00FF",  "Symbol": "...",  "IsEnabled": 1,  "IsSearchable": 0,  "IsGroup": 1,  "IsChildrenOrdered": 1,  "OrderNo": 3,  "Aspects": [  {  "Id": 8,  "Title": "Hemmelighed 1",  "Description": "...",  "Value": "Hemmelighed 1",  "Color": "772277",  "OrderNo": 1,  "Phases": [  {  "Id": 1,  "Title": "Hint",  "Description": "...",  "OrderNo": 1  },  {  "Id": 2,  "Title": "Berøring",  "Description": "...",  "OrderNo": 2  },  {  "Id": 3,  "Title": "Udfoldelse",  "Description": "...",  "OrderNo": 3  }  ]  },  {  "Id": 9,  "Title": "Hemmelighed 2",  "Description": "...",  "Value": "Hemmelighed 2",  "Color": "55FFFF",  "Searchstring": "",  "OrderNo": 2,  "Phases": [  {  "Id": 4,  "Title": "Hint",  "Description": "...",  "OrderNo": 1  },  {  "Id": 5,  "Title": "Berøring",  "Description": "...",  "OrderNo": 2  },  {  "Id": 6,  "Title": "Udfoldelse",  "Description": "...",  "OrderNo": 3  }  ]  }  ]  } |
| Section | Account | Work | Section  ⮡ Section (children)  ⮡ AspectOccurrence | {  "Sections": [  {  "Id": 1,  "SectionLevel": 4,  "StartPos": 0,  "Length": 0,  "Entry": 1,  "Position": 1,  "AspectOccurrence": [  {  "Aspect": 8,  "Phase": 1  },  {  "Aspect": 1  },  {  "Aspect": 2  }  ],  "ChildSections": [  {  "Id": 2,  "SectionLevel": 5,  "StartPos": 0,  "Length": 100,  "Entry": 1,  "Position": 1,  "ChildSections": [  {  "Id": 3,  "SectionLevel": 6,  "StartPos": 0,  "Length": 20,  "Entry": 1,  "Position": 1  },  {  "Id": 4,  "SectionLevel": 6,  "StartPos": 0,  "Length": 30,  "Entry": 1,  "Position": 2  }  ]  },  {  "Id": 5,  "SectionLevel": 5,  "StartPos": 0,  "Length": 200,  "Entry": 1,  "Position": 2  },  {  "Id": 6,  } | "SectionLevel": 5,  "StartPos": 0,  "Length": 300,  "Entry": 1,  "Position": 3  },  {  "Id": 7,  "SectionLevel": 5,  "StartPos": 0,  "Length": 0,  "Entry": 1,  "Position": 4  }  ]  },  {  "Id": 8,  "SectionLevel": 4,  "StartPos": 0,  "Length": 0,  "Entry": 2,  "Position": 2,  "AspectOccurrence": [  {  "Aspect": 8,  "Phase": 2  }  ]  },  {  "Id": 9,  "SectionLevel": 4,  "StartPos": 0,  "Length": 0,  "Entry": 3,  "Position": 3  },  {  "Id": 10,  "SectionLevel": 4,  "StartPos": 0,  "Length": 0,  "Entry": 4,  "Position": 4,  "AspectOccurrence": [  {  "Aspect": 8,  "Phase": 3  }  ]  }  ] |
| Entry | Account | Work | Entry  ⮡ Entry version | {  "entries": [  {  "Id": 1,  "Subject": "...",  "OrderNo": 1,  "Extract": "Ellen er sent på den. Hun haster igennem den lyse atriumgård på Ny Odenses IT universitet."  },  {  "Id": 2,  "Subject": "...",  "OrderNo": 2,  "Extract": "Hele vejen rundt om plejehjemmet er der anlagt en stor have som igen er omkranset af høje popler, hvis toppe svajer roligt i vinden."  },  {  "Id": 3,  "Subject": "...",  "OrderNo": 3,  "Extract": "Det bliver en god dag. Man er ulasteligt klædt og der er ikke en plet at se. Robert står ved bordet med champagneglas, og i hænderne holder han en Krug Clos d'Ambonnay, klar til at slippe proppen, når gæsterne kommer ud i pausen. Da gæsterne træder ud, fyldes rummet med røg."  },  {  "Id": 4,  "Subject": "...",  "OrderNo": 4,  "Extract": "I Robert tændes der et direktiv, hans eget. Han forlader sin plads foran terrassedørene"  }  ]  } | |

### Blobs

|  |  |  |  |
| --- | --- | --- | --- |
| Blob | Container | type | name |
| Entry | Account | Text files | Id |
| Recording | Account | Binary files | Id |

### Files

|  |  |  |
| --- | --- | --- |
| File |  |  |
|  | Exported files… |  |
|  | Imported files… |  |

**NOTE**; aspects are linked to sections always unless they are searchable in which case thet are still linked using some kind of section. That information is added to the html markup, so it is easy to use e.g. jqueery to identify the marks and from that info draw/highlight the relations/information

**NOTE**: We can use Saxon XSLT 3.0 to transform json to different structures so we can e.g. list users from accounts or list aspect occurrences from sections: It is .NET Framework 3.5 and free for non commercial. Price is otherwise 1.200 pounds for a site-license. Cheeper alternative is to convert json to xml and then just use XSLT 2.0 ( [Converting between JSON and XML with Json.NET](http://james.newtonking.com/projects/json/help/index.html?topic=html/ConvertingJSONandXML.htm))

**NOTE**: Data is almost isolated in works. There are few places where this is not true

* Links bentween entries in different works - include other work's id
* User access to works in other accounts - include foreign account's id

This allows for all ids for entities within works to have relatively unique ids, that will seldom exceed 1000s.

You simply have a local number roll to pick from, which is part of your model - e.g. the account or the Work.

Splitter:

<https://www.jqwidgets.com/jquery-widgets-demo/demos/jqxsplitter/index.htm#demos/jqxsplitter/defaultfunctionality.htm>

# Detailed requirements

|  |  |  |
| --- | --- | --- |
| Legend | Requirement | Description |
| 🖌 | *Graphical* | Graphical content, panels, swiping etc. |
| 🗚 | *Entry* | Length of text in entry fields, formatting etc. |
| 🗩 | *Validation* | Validation of input or actions. |
| 🡺 | *Navigational* | How the user moves around between different content. |
| 📋 | *Rules* | Rules. |
| 🗲**A#** | *Actions* | Changes to the data and operations like sending emails etc.  A number helps finding the action in the activity diagram. |
| 🗄 | *Data* | How to store information. |
| ✔ | *Implementation* | Implementation of functionality |
| 🛠 | *Technical* | Implementation and compatibility with machines and sub systems. |
| 🖧 | *Integration* | How to collaborate with other systems or services. |
| ⦸ | *Security* | Authentication etc. |

# Construction

## Note taking integration

|  |  |
| --- | --- |
| **Diigo**  User: [public@weloom.dk](mailto:public@weloom.dk) | <https://www.diigo.com/api_dev/docs#section-overview>  App Name: MindMyPen  Description: Tool for authors and technical writers. THey might need to import notes from Diigo  **API key: bd6db5d930065a81** |
| **Evernote**  <https://www.evernote.com/client/web>  Devusername: d  User: public@weloom.dk | <https://dev.evernote.com/doc/#reference>  Here is your Evernote API Key  It comes in two parts: a Consumer Key and a Consumer Secret. Both are required to authenticate with the Evernote API and receive an authentication token. Your Consumer Secret should be kept private and not shared.  **Consumer Key: d-1386**  **Consumer Secret: 6859efcddc657355**  Both of these values will be sent to you via email. |
| **Hypothesis**  <https://web.hypothes.is/>  User: md@weloom.dk | <https://h.readthedocs.io/en/latest/api-reference/v1/>  **API toke: 6879-zszPTh-pWSpXQG7-LG9aWQcWpmN5XP4GbeMNCc-Fxss**  host: virtserver.swaggerhub.com  basePath: /Weloom/Hypothesis/1.0.0  schemes: https |