# MVP - Onlive

*Functional documentation - May 2020 - v5 - Smartup Zero*

Contains:

1. MoSCoW method description
2. Product Description
3. Introduction to the UX design
4. Concept Definitions
5. EPICS for Minimal Viable Product


## Abstract:

In this document you'll find an overview of the research results into the User Experience and the definition for the Minimal Viable Product of Onlive. For this research we used the MoSCoW method to help us prioritize and define. This document should give developers and designers enough fundament to map out their activities to build Onlive collaboratively.

# MoSCoW Method:

The MoSCoW method is a prioritization technique used in management, business analysis, project management, and software development to reach a common understanding with stakeholders on the importance they place on the delivery of each requirement; it is also known as MoSCoW prioritization or MoSCoW analysis.

All requirements are important, but they are prioritized to deliver the greatest and most immediate business benefits early. Developers will initially try to deliver all the *Must have*, *Should have* and *Could have* requirements but the *Should* and *Could* requirements will be the first to be removed if the delivery timescale looks threatened.

The plain English meaning of the prioritization categories has value in getting customers to better understand the impact of setting a priority, compared to alternatives like High, Medium and Low.

The categories are typically understood as:

**Must have**

Requirements labelled as *Must have* are critical to the current delivery timebox in order for it to be a success. If even one *Must have* requirement is not included, the project delivery should be considered a failure (note: requirements can be downgraded from *Must have*, by agreement with all relevant stakeholders; for example, when new requirements are deemed more important). *MUST* can also be considered an acronym for the Minimum Usable Subset.

**Should have**

Requirements labelled as *Should have* are important but not necessary for delivery in the current delivery timebox. While *Should have* requirements can be as important as *Must have*, they are often not as time-critical or there may be another way to satisfy the requirement so that it can be held back until a future delivery timebox.

**Could have**

Requirements labelled as *Could have* are desirable but not necessary and could improve the user experience or customer satisfaction for a little development cost. These will typically be included if time and resources permit.

**Won't have (this time)**

Requirements labelled as *Won't have*, have been agreed by stakeholders as the least-critical, lowest-payback items, or not appropriate at that time. As a result, *Won't have* requirements are not planned into the schedule for the next delivery timebox. *Won't have* requirements are either dropped or reconsidered for inclusion in a later timebox. (Note: occasionally the term *Would like to have* is used; however, that usage is incorrect, as this last priority is clearly stating something is outside the scope of delivery).

# Product description

## What does Onlive do (in its adult form)?

**1: Grouping people within the dynamics of physical shared spaces and the common resources available there, e.g. a group of people that:**

- Are in the same room
- Work in the same office
- Shop in the same store
- **Are at the same event** VERSION 1.0
- Live in the same flat
- Are on the same public square
- Live in the same street

**2: Administration of the mutual exchange of relevant information between humans, computers and group hosts that are the stakeholders in that physical context.**

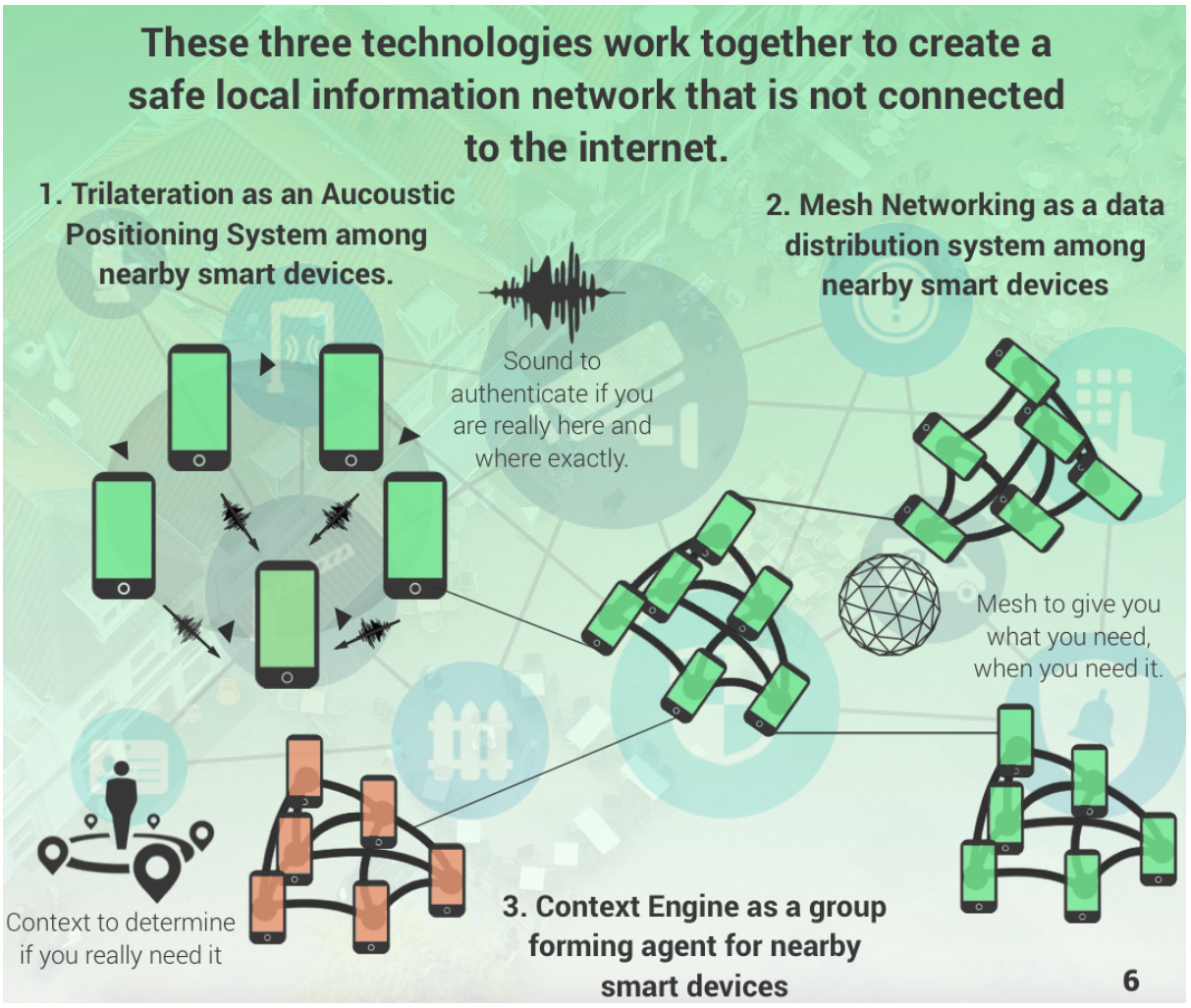
- Mesh-networking to distribute data among group members (and their smart devices).
    - POC does not need a mesh implemented and operationable but has a 'normal' client/server set up that simulates mesh. That way we can implement mesh in a later stage. Mesh is on the R&D roadmap.
- Context rules to determine relevance for the group (and their smart devices)
    - Fully functioning.
- Plugin runtime to execute web applications inside the groups
    - Fully functioning.

**3: Facilitating functionality for:**

A. Disaster control services (Onlive works where other communication infrastructrures fail).
    i. Virus control, i.e
        1. Group control and monitoring, e.g:
        2. Warning system for crowded groups
        3. Heatmap for local area's around the user
        4. Entree policy for bars and clubs
        5. Social distance information
    ii. Post diaster communications, i.e
        1. Real time information from communicities in need of help
        2. Organizational tools to share resources in communities

3. Sharing social information about loses and founds of people and goods.

B. Collaboratively enriching information:
    i. Group organization and planning, e.g:
        1. Filling in a poll at a convention
        2. Working in the same document in a meeting.
        3. Report local incident (e.g. a bike being stolen)
C. A better collective experience in our physical spaces.
    i. Using local available digital services, e.g:
        1. Local payment system in a restaurant
        2. City guide system for tourists in a city
    ii. Using local available smart devices, e.g:
        1. Controlling the available smart tv in the room
        2. Controlling the music system in a bar (jukebox)
        3. Using local available IoT data (smog alert system)
    iii. Sharing local common resources, e.g:
        1. Local car sharing
        2. Local food sharing
        3. Local energy sharing
D. Connecting demand and supply of services between people and organizations
    i. Sharing supply, e.g:
        1. Offer your time to help somebody in your neighbourhood
        2. Offer professional services to organizations/people in your neighbourhood
        3. Offer commercial opportunity to people in a specific physical context
    ii. Sharing demand, e.g:
        1. Request help with getting groceries
        2. Request discussion about an idea
        3. Request the best commercial deals from your local businesses.


These three technologies work together to create a safe local information network that is not connected to the internet.
1. Trilateration as an Aucoustic Positioning System among nearby smart devices.
Sound to authenticate if you are really here and where exactly.
2. Mesh Networking as a data distribution system among nearby smart devices
Mesh to give you what you need, when you need it.
Context to determine if you really need it
3. Context Engine as a group forming agent for nearby smart devices
6

# Introduction to the UX design

To teach and experience the fun and usability of Onlive we've designed the UX based on the physical and social laws we are familiar with when we are playing a game of cards with a group of people. Going through this document you'll recognize all kinds of basic principles we all know from card games that involve activities such as trading, collecting and using cards but also administrative activities such as deck building.

Specifically, we want to use the card game dynamics that users, prior to group participation, carefully consider which cards they want to use (publish), collect (stacking) and exchange (sharing) in that group. The cards that you know you will need in a specific group you prepare and order, prior to entering a group.

e.g. Stacking the cards you are going to need when you know you are going to a networking event.

Another great way of thinking about this is to think of the process of packing your backpack. You know what you are going to do that day, so you are preparing the stuff you know you will need for sure.

It enabled us to comply with our goal to build a system that evokes fewer interactions with our smartphone and stimulate more human interaction at the moments we are around people we need to work/live together with. Also, it allows us to make it very transparent who the Card is from, what kind of data is used and stored and where the card originates from.

# Concept definitions

**Plugin:**

Plugins are software components that create or edit Cards.

**Card:**

A small, recognizable, collectable and exchangeable digital object holding data that is managed by a plugin. Includes information about the:

1. Owner
    i. The owner is the person who created the card and first published it inside a group.
2. Plugin:
    i. The web application facilitating the functionalities concerning creation, use and processing of the card type.
3. A content preview:
    i. A visual reference to the content of the card that allow users to quickly identify its function..
4. A phase indication:
    i. Indicates whether a card is active or passive (useful or inactive)
5. A storage indication:
    i. Indicates whether the card may be saved by the group members.
6. Publication indication
    i. Shows if a card is currently published on the stage.

**Stage:**

The environment in which cards can be shared and used by other members of the group.

**Hand:**

An environment where a single card can be viewed and where all essential direct interactions with the card happen:

- **Save (swipe it to your body):**

  Keeping the card in your ‘saved cards’

- **Hold (swipe it to your right):**

  Keeping the card ‘nearby’ so you can find and use the card at any moment while participating in the group.

- **Publish (swipe it up)**

  Drop the card on the stage.

- **Drop: (swipe it to your left)**

  Send it back where it came from.

**Card Type:**

The type of card that defines an initial understanding about the function of a card to the end-user.

**A stack:**

A collection of 0 to ... cards that are connected to each other under a tag.

**Copy / duplicate:**

Creating a copy (new instance) of an existing card and use it as a template to edit and publish as a new card.

**Export:**

Exporting the data of a card for usage with specialized applications outside of the platform.

**Deck / stack building:**

Deliberately stacking (preparing) cards with the goal to share/use them collectively or to have them available when participating in a group participation.

**Filter:**

Applying metadata of cards to filter out irrelevant cards to the user.

**Search:**

Actively searching for specific card (types)  by supplying text input, whether or not combined with the application of filters.

**Context Group:**

A collection of users that all share the defined context, registered/configured by a group host.

**Onlive group:**

The permanent available group that is publicly accessible to everyone that is near enough to hear a subsonic sound emitted between smart devices. Hosted by Onlive.

**Group notification:**

These are the notifications from within and outside the group, that you're actively a part of, giving information about important events in the group.

**Host:**

The host is the owner of a group that determines who can participate and how users can interact in their group. Moderation, editors etc.

**Plugin overlay:**

Omnipresent Onlive functionalities remain visible while using a plugin.

**Savable card:**

After publication, these cards can be stored by other group members.

**Non-savable card:**

After publication, these cards can’t be stored by other group members. They can only be viewed and used while being part of that specific group.

**Retraction:**

A user can always withdraw a card from the stage. In this case all ongoing operations taking place with the card by other group members are suspended immediately. A saved card remains available and can’t be retracted by the owner.

**Badge:**

A badge can be awarded for validated information about the user. It is a mark (icon) that can be found within the profile of a user, and can be traced back to the accrediting party.

## User types / roles:

Types:

1. Individual user (group member)
2. Group host / owner

Roles:

1. Sharing
2. Creating
3. Contributing
4. Collecting
5. Lurking

6. Facilitating
7. Moderation

**The interactive mockup to guide this functional description can be found here:**
**https://xd.adobe.com/view/6aac4500-8047-4aac-476e-7dc994e21ae2-1c9d/**

# 1: Card Home: your saved stacks / cards

## Mockup view:


## Module goal:

To provide an overview to the end user of it's stored cards and stacks. From this overview stacks are formed which can be used within groups. Next to 'Profile', the Card Home is the only module users can use without being part of a group.

## UX framework (meaning)

Within *Card Home* a user can find all cards that were saved in the groups they recently were part of. All cards are saved as a single object, but can also be linked together in stacks. Stacks (piles) stick to each other based on associated information and functionalities so that they can easily  be found, shared, and can be enriched within the groups. Both individual cards and stacks must be able to be easily found with smart filter system and a strong search protocol.

**Keeping your saved cards in order:**
All cards that the user saves when participating in groups, end up in saved cards. The user is challenged to assess the cards and place them in stacks. For example, meaningful cards are kept in new or existing stacks, saves or deletes still loose.

**Specialization module:**
The saved stacks / card module is not intended to function as the primary repository of data and content. It is not your Windows Explorer, Finder or other relatives. The saved cards module should facilitate Onlive users a fun, usefull and satisfying environment to prepare and process cards for usage within groups and/or stacks.

**Export:**
Cards with contents that need to be edited/processed by external applications outside the platform and they are no longer relevant to be useful for the end use to use in Onlive groups, should be permanently exported. If a card has been exported and edited with external applications it may find it'sway back to the platform by being part of a new card.

Important to note is that we should consider the function to archive cards, so exported and deleted information within cards is still traceable on a later moment.

**Duplicate from original**
Cards saved from the stacks / cards can also be used for the re-sharing information, the re-use of functionalities, and the re-use of the card structure  (template). The latter requires the ability to easily copy an existing card so the content or structure can be modified.

## User stories / scenarios

1. Add card to stack (S)
2. Delete from card stack (S)
3. Make new stack (S)
4. Search cards (S)

5. Search stacks (S)
6. Filter stacks / card (S)
7. View card (M)
8. Duplicate card (S)
9. Open card (M)
10. Organize new cards (S)
11. Export plugin content? (C)
12. Export card (C)
13. Import card? (C)
14. Publish card (M)
15. Archive card (C)
16. Migrate cards between accounts (C)

## Epics MVP

### View card

A User must be able to choose a card from the Card home

### Open card

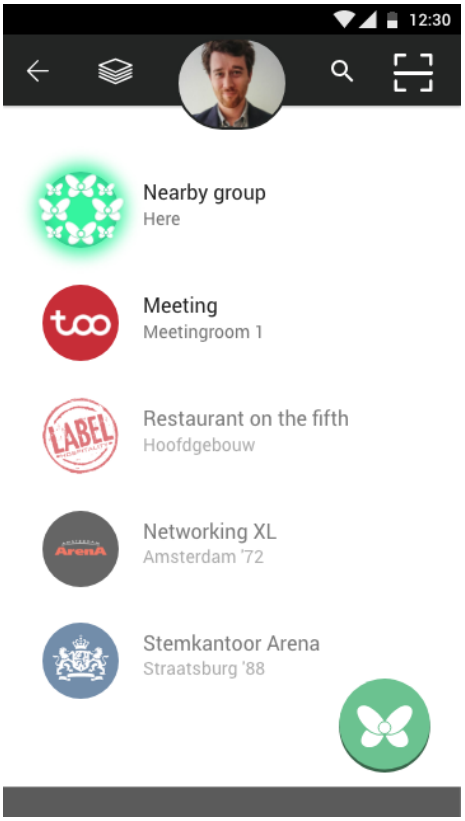
User must be able to open selected card

### Publish card

User can publish selected card for available groups.

# 2. Groups

## Mockup view:




## Module goal

Providing insight into the available context groups you can participate in or have been participating in. Group Notifications should be visible here.

## UX framework (meaning)

This environment is the default module of the app. Onlive ensures that the most relevant groups are visible to the user. There is one Onlive group that is always present, that you can use to interact with users in your immediate surroundings.

## User stories / scenarios

1. Join group (go to stage) (M)
2. Load group (qr scan) (M)
3. Hide / block group (W)

    4. Leave group (M)
5. Find group (W)
6. Sort group (C)
7. Make group (W)
8. Navigate to group (W)
9. Group notification (S)
10. Go to profile (M)
11. Go to Card Home (M)

## Epics MVP

### Join group (go to stage) (M)

User can navigate between groups available

### Load group (QR scan) (M)

User can scan an QR code at a location and load a group in the list

### Leave group (M)

User leaves the group and is no longer visible in the list.

### Go to profile (M)

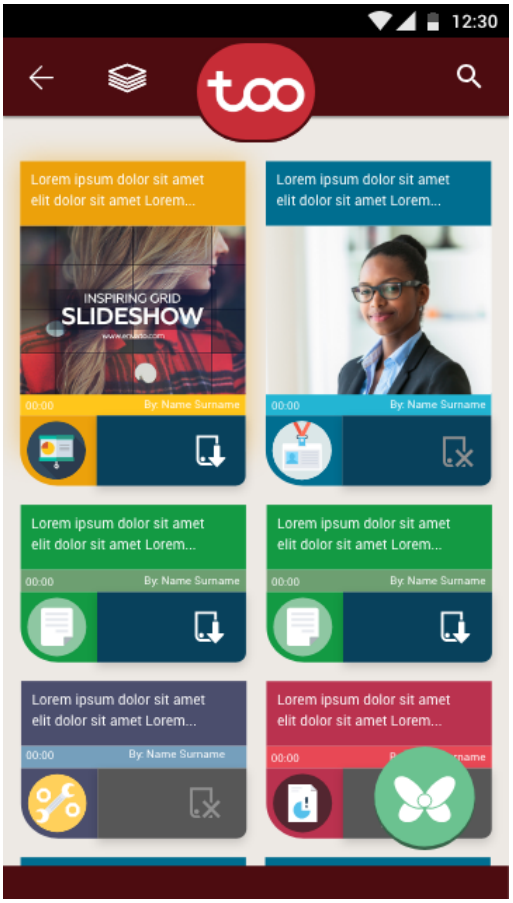
User navigates to personal profile

### Go to Card Home (M)

User navigates to Card Home

# 3. Stage

## Mockup view:




## Module goal

To offer group members a common place for published cards inside their groups.

## UX framework (meaning)

Within a physical space, the Onlive stage is the virtual table where all published cards (by people and machines) are available for group members to interact with. The group host is responsible for what kind of cards are allowed on the stage. Group members can use the cards on the stage quickly use, exchange, process information (together) and make use of the facilities within the physical space.

On the stage Onlive optimizes the positioning of cards and for each individual user. Furthermore, Onlive offers powerful search and filtering options to allow users to quickly find the cards they need.

## User stories / scenarios

1. Take card in hand (pick it up from the stage) (M)
2. See the group members that can also see and use these cards (M)
3. Dynamic positioning change cards based on relevancy (C)
4. Select/create New card for stage (M)
5. Go to Hand (M)
6. Filter cards (M)
7. Search cards (S)
8. Change identity (account) (C)
9. General notifications (C)
10. Personal notification (C)
11. Group ending / exit (M)
12. Hide card (C)
13. Make stack (S)
14. Host (C)
    a. end group
    b. Kick user / organization
    c. Delete card
    d. Archive card
    e. Make notification to group
    f. Make notification to user
    g. Make stacks on stage
    h. Change card priority
    i. Lock stage
15. Pin card to stage (S)
16. Retract all cards (S)

## Epics MVP

### Take card in hand (M)

User takes a card from the stage in hand

### See group members that can see and use the same cards (M)

User can see who are present within the group

### Select/create New card for stage (M)

User publishes new and existing cards in the stage

### Go to Hand (M)

User navigates to  hand

### Filter cards (M)

User filters cards within the stage based on modification date and card type

### Group ending / exit (M)

User leaves the group

# 4. Hand

## Mockup view:


## Module goal

An environment in which a user can perform basic personal interactions with a card.

## UX framework (meaning)

The stage is an area that is accessible to everyone in the group. The hand is the module that offers the user a shielded "safe" environment. In the hand, the user is actively interacting with a card. That card of the matter can come off the stage, picked from your Card Home or be a freshly created new card, created on the fly using a plugin.

With a card in your hand you can perform the following activities:

- **Save (swipe it below):**

  Keeping the card in your ‘saved cards’

- **Hold (swipe it to your right):**

  Keeping the card ‘nearby’ so you can find and use the card at any moment while participating in the group.

- **Publish (swipe it up)**

  Drop the card on the stage.

- **Drop: (swipe it to your left)**

  Send it back where it came from.

## User stories / scenarios

1. Save card (M)
2. Hold card (M)
3. Browse stack cards (S)
4. Make card (M)
5. Make stack (W)
    a. Request stack
6. Drop (M)
7. Publish (M)
8. Open card (M)
9. View metadata (back of card) (S)
    a. See all plugin card info
    b. See all card information source
    c. See all card history
10. Reach hold limit (C)
11. Duplicate card (C)
12. Group ending / exit (closing processes) (M)
13. Card retraction (M)
14. Retract card (M)

## Epics MVP

### Save card (M)

User saves a card to Card Home

### Make card (M)

Create a new card using available plugins in the group

### Drop Card (M)

User returns the card without adjustments to where it came from

### Publish Card (M)

User publishes new, enriched or edited card to the stage

### Open card (M)

User accesses the contents of a card triggering  plugin runtime

### Group ending / exit (closing processes) (M)

User receives a notification when a group becomes unavailable.

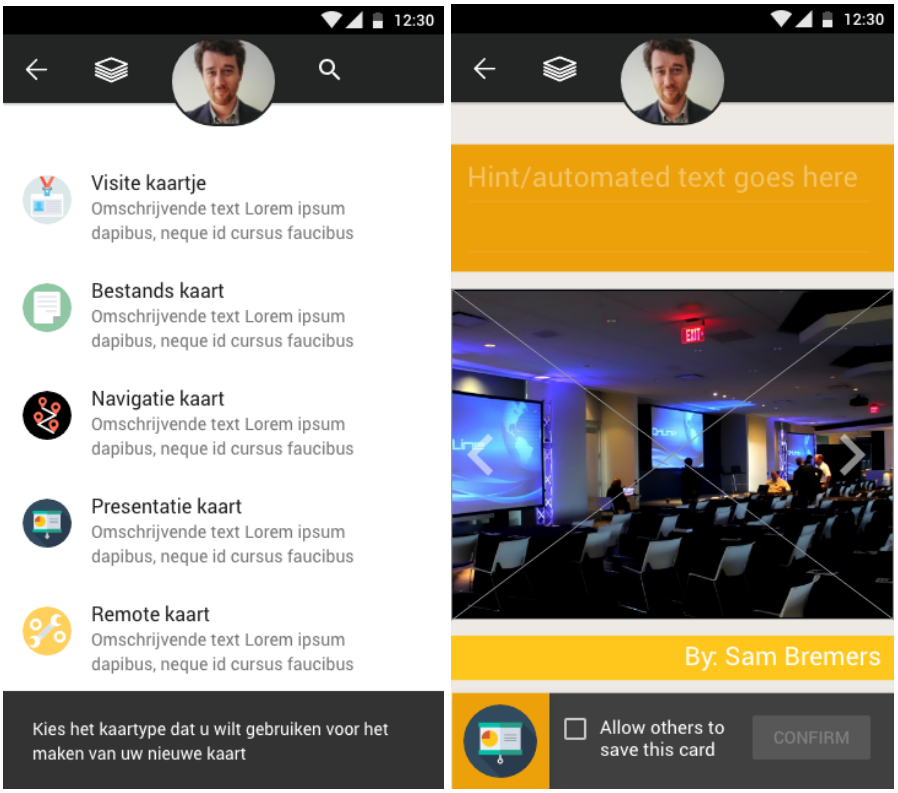
### Card retraction (M)

User gets notification when a card is unavailable

### Retract card (M)

User pulls published cards back from the stage and all hands

# 5. Create card wizard




## Module goal

Guiding the user step by step, as efficiently as possible, through the process of creating  a card.

## UX framework (meaning)

Cards can be created from a specific group environment using plugins that are permitted there. If a user wants to make a card he must complete a number of steps (wizard). These steps are partly facilitated by Onlive and partly by the plugin.

First the user chooses what type of card he wants to create. There is  list that presents him the options available in the group he is participating in. The user then goes through a certain number of steps to facilitated by the plugin developers. These steps are defined by the functionality of the plugin. After these steps, the card includes information, holds the proper content and configuration and is able to perform a functionality that is defined by the plugin. The user then adds a title, chooses a thumbnail and determines if the card should be able to be saved/copied. Then the card is ready, in the Hand and can be Published, Saved, Hold or Dropped.

## User stories / scenarios

1. Make new card using 3rd party plugin (full process)
    a. Select card type (M)
    b. Create content using plugin steps (M)
    c. Choose card preview (M)
    d. Choose title card (M)
    e. Choose saveable / unsaveable (M)
    f. Preview card (in hand) (M)
2. Make new card using 3d party automatically (full process)
    a. Select card type (M)
    b. Create content using plugin steps (M)
    c. Choose card preview (M)
    d. Choose title card (M)
    e. Choose saveable / unsaveable (M)
    f. Preview card (in hand) (M)
3. Edit duplicated card from Home: Saved Cards (S)

# Epics MVP

## Make new card manually

You create quickly a new card through simple steps.

**Select card type (M)**

User chooses a plugin from a list to make a card

**Create content using plugin steps (M)**

User made the contents of the card in the vicinity of the plugin

**Choose card preview (M)**

User chooses a visual preview of the card content on the card

**Choose title card (M)**

User type a title on the card or customize it to

**Choose saveable / unsaveable (M)**

User chooses stores may be on a card

**Preview card (in hand) (M)**

User rated the card as a whole

## Make new automatic card (full process)

The plugin automates steps when creating a new card

**Select card type (M)**

Plugin is selected automatically for the user

**Create content using plugin steps (M)**

Contents of the card is automatically created.

**Choose card preview (M)**

The visual preview of the content is automatically generated

**Choose title card (M)**

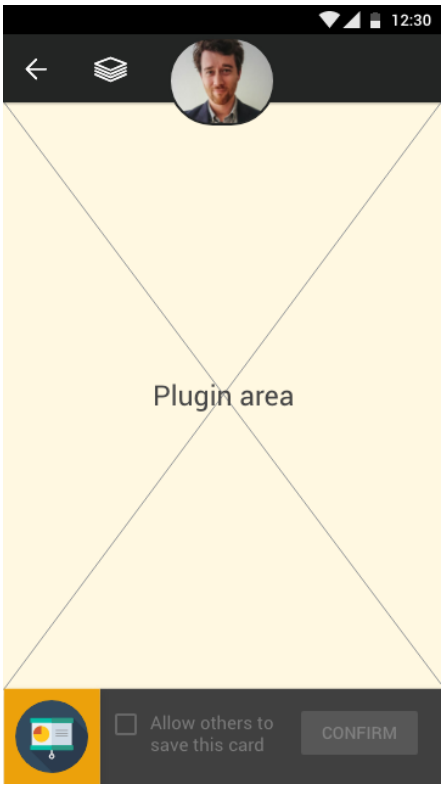
The plugin automatically generates a title for the card

**Choose saveable / unsaveable (M)**

The plugin automatically selects whether the card can store may be**n**

# 6. Overlay Plugin

## Mockup overview:




## Module goal

A recognizable set of (navigation) features that the user can use while inside a plugin interface.

## UX framework (meaning)

Users use Onlive in ever changing and dynamic circumstances. The process of card creation can be long. The user should be able to return to the hand and/or stage quickly if his social situation requires that.

## User stories / scenarios

1. Return to onlive mid position (M)

        a. Plugin process stops
    b. Plugin process pauses
2. Group notification (C)
3. Card mid retraction process (M)
4. Group ending / ended (S)
    a. card savable
    b. Process terminated / pauzed
5. Support request (W)

# Epics MVP

## Return to onlive mid position (M)

User returns to the stage or the hand

### Plugin process stops

User returns to onlive and all plugin processes are interrupted

### Plugin process pauses

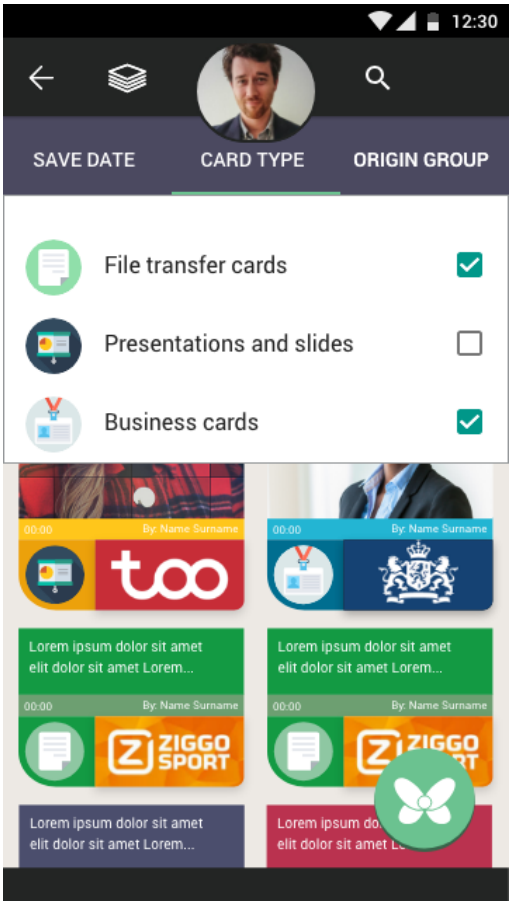
User returns to onlive and plug process are paused

## Card mid retraction process (M)

The plugin process stops immediately and the user sees that the card is retracted

# 7. Search & Filter:

## Mockup view:




## Module goal

Finding cards and stacks.

## UX framework (meaning)

The user can use two autocomplete instant search engines that operate on the input of text. One focuses on the personal collection of cards and used within the Card Home. The other focuses on search within a specific stage.

In addition to searching cards can also be filtered separately or simultaneously. We can distinguish the following search tags and associated filters.

1. Titles

        2. Date of saving
3. Group of origin
4. Date of modification
5. Publisher
6. Plugin type

## User stories / scenarios

1. Search cards within stage (s)
    a. On titles
    b. Time of publishing
    c. Date of modification
    d. Group origin
    e. Card issuer
    f. Card author
    g. Plugin types
    h. Organizations / individuals
2. Filter / sort cards in stage (M)
    a. On tracks (alphabetical) (C)
    b. Time of publishing (new / old) (C)
    c. Edited (M)
    d. Card issuer (s)
        i. Organization / Person (W)
    e. Card Types (M)
    f. Organizations / individuals (W)
3. Search cards within Card Home (S)
    a. On tracks (C)
    b. Date of storage (new / old) (M)
    c. Group (M)
    d. Card issuer (s)
        i. Organization / person (W)
    e. Card types (M)
    f. Organization / person (W)
4. Filter / sort cards Card Home (M)
    a. title alphabetically
    b. Date of storage (new / old)
    c. Edited
    d. Group
    e. card issuer
    f. card author
    g. plugin types
    h. Organizations / individuals

# Epics MVP

## Filter / sort cards in stage (M)

User filters and / or arranges the supply of cards on the stage

**Edited (M)**

filters user / arranges cards based on processing date

**Card Types (M)**

filters user / ordernt based on selected card types

## Filter tickets Card Home (M)

User filters and / or arranges the cards within their preserved collection

**Date of storage (new / old) (M)**

filters user / orders based on the date they were acquired

**Group of origin (M)**

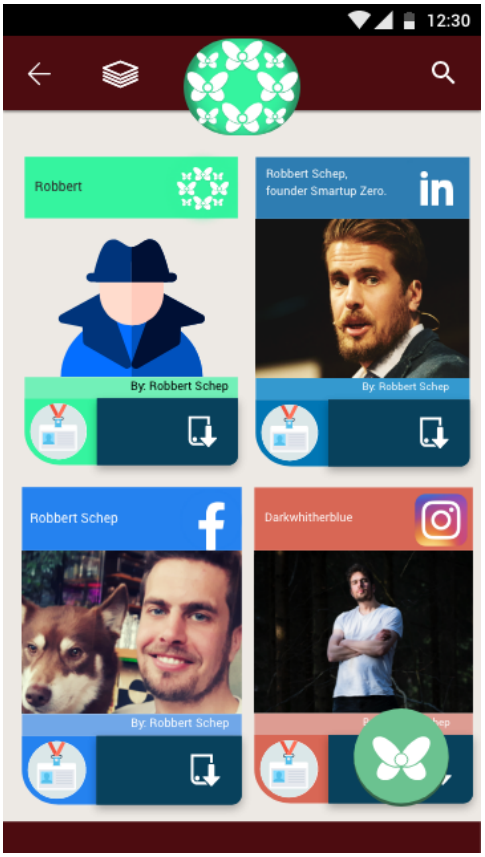
filters user / orders on the basis of the groups in which they were obtained

**Card types (M)**

filters user / orders based on existing card types

# 8. Profile and identity

## Mockup view:




## Module goal

Create, edit, control and manage under what privacy conditions your participate in groups.

## UX framework (meaning)

Onlive is a network for the real world and in the real world you can't be invisible. Especially when you are in a group that needs to work together to accomplish or experience something, the presence of your body, identity and purpose are vital for the level of trust there is among your group. The less we know about each other, the less effective we are as a group. Trust is vital for effective collaborations. Therefore, Onlive strives to be a tool to create trust among group members by implementing a digital equivalent of our physical presence. This means your presence and participation in groups can never be completely anonymous. However, Onlive lets you control exactly how you present yourself the groups you participate in and what

information you want to share about yourself at any given moment. This is one of the core goals of the Onlive Cards. However, this all starts with the login process.  Onlive is a network that you have to log in to.  You do that with choosing your ID card.

## UX specification

There are two types of ID cards:

1. Your Onlive ID Card is like your passport. Everybody needs to have one to be able to use Onlive. When you create your account you will automatically get one. It's attached to a unique Onlive ID number (uuid) that you broadcast as a request to the Onlive server to log in. In default, it lets you join other groups anonymously. This means that other group members only see your username. However, interactions with other cards and plugins are always connected to your Onlive ID number, and thus to your account. Just like in the real world you can be anonymous to look around, but you can't be invisible if you want to participate in a group.
2. Although your Onlive ID card is a great way of safely and quickly 'look around' what's going on in the relevant groups, it's better to log in with one of your Role Cards. A Role card is a version of your Onlive ID-card but then with specific privacy settings. It determines what kind of personal information you are sharing and how you can use the network. In essence, you are choosing the role you want to play in a group. Depending on the purpose of your visit, e.g. work, leisure, shopping, traveling or simply browsing, you can control different role cards that help you gain control over your own data and how you share that with the world around you. Also, it will help the Onlive system selecting the relevant groups for you, to optimize the effective experience.

## Research:

For privacy related questions about how we deal with the users identity in a group and it’s interaction with cards and plugins, it’s important to understand we built Onlive to support people in their daily lives when they need to interact with their varying surroundings. We want to optimize but minimize those interactions because we want to reduce redundant screen time and increase quality face-to-face talk.

The goal is to cooperate and communicate with those in our immediate world, in such a way that we become better in fixing our world. If you want to build a digital system for the physical world you have to apply certain rules of the physical world when you design the system. One of the key rules in social group behaviour is that the more trust there is, the more effective we communicate.

*"When trust is present, people step forward and do their best work, together, efficiently. They align around a common purpose, take risks, think out of the box, have each other's backs, and communicate openly and honestly. When trust is absent, people jockey for position, hoard information, play it safe, and talk about— rather than to—one another".* - Dennis Reina, PhD Michelle Reina, PhD David Hudnut, MIA, Why Trust Is Critical to Team Success, 2017

Onlive has to become and stay a tool that people can trust like a strong spade that everyday helps people to work their ground. Something to rely on. Something you can just use and you don't have to worry about any scam or failure. To establish trust in groups we've mimicked our natural group conventions as close as possible when we made some key UX decisions. The most direct evidence of such a choice is that we follow the natural law of matter that dictates that you can't be invisible when you are participating in a group. Therefore with Onlive you can't be anonymous when you participate in a group.

This sensitive and complex matter seems to be an issue that can be handled in a later stage when people are actually using the platform but in present days it has never been more important that you have to be clear and transparent about your privacy policies from the beginning. Onlive wants to do things different than traditional social media and privacy is one of those areas where we can establish that difference.

To address this issue head on we introduce the Onlive passport. The premise of this idea is, again, mimicked from our physical world. When you travel to another country you bring your passport. When you want to use Onlive you have to have an Onlive ID card.

The Onlive ID card is made when you register as a user for the first time. It's basically a quick registration process we are familiar with when we use other online platforms.

- Email (Mandatory and to be verified)
- Phone number (Mandatory and to be verified)
- Name (Mandatory and not verifiable)
- Username (Mandatory and not verifiable)
- Photo (Not mandatory and not verifiable

After this registration process Onlive will assign a unique number in the form of an UUID. This UUID is connected to your personal information and together they form your Onlive passport that allows you to enter and participate in groups.

Regarding privacy issues we have to address two specific situations:

1. What happens when you enter a group but don't interact with any of the available cards.
2. What happens when you enter a group and interact with one or more cards.

We separate these two situations because Onlive basically just facilitates the existence of groups. What happens in the group is mostly determined by the group host and the available plugins.

In general you could say that Onlive only keeps track of the groups being formed and the users that ENTER the group but does not keep track of the INTERACTIONS that happen inside the group.

Interactions with cards are being tracked by card owners. Interactions with plugins are being tracked by plugin owners.

*What happens when you enter a group but don't interact with any of the available cards?*

- In the Onlive group:
- Your UUID will be used as a sign-in to the group and will display your Username and Photo in the attendees list for other people to see. Notice that you your Username does not have to be your real name.
    - your UUID will be registered in the Onlive backend, connected to the time and place of the groups existence. The Onlive backend only saves the UUID for 24 hours and than deletes all registrations. The Onlive backend will ONLY save the UUID and not the personal information that belong to that UUID.
        i. This information will never be given to third parties unless that information can help verified government organisations solving criminal activities.

- What happens when you enter a Context group depends on the mandatory privacy decisions the group host made when creating the group. We imagine that there will be two options group host can choose from:

A. Standard: the same privacy rules as for the Onlive group
B. Custom: the sign-in process requires extra permissions of users depended on the Identity plugin the group host choices .

For example: a group host creates a group for his networking event. He wants his visitors to share more information about themself than just a username and a

picture. In order to do that he has to install an Identity Plugin for his group that complies with his identity demands. In this case he chooses the business card plugin that demands from users to sign in with their business card. When the user enters the group that plugin will ask the user to publish his business card to the stage. If the user does not have a business card he can create one very quickly with the business card plugin.

This way we can foresee different kinds of identity plugins that generate different kinds of sign-in cards. This is a smart and controllable way of creating transparent situations regarding the privacy of users. They are either signing in with their Onive passport or are asked to hand in a different form of identification that is chosen by the groups host.

*What happens when you enter a group and interact with one or more cards*.

The general rule is that your Onlive passport will always be attached to your interactions with cards. It does depend if a card is marked by the owner as savable, collectable, reusable or re-publishable what exactly happens but in general this is what happens:

- If you create a new card, your Onlive passport is registered als 'Owner' of the card (there can only be one owner of a card)
- If you save a card of somebody else, your Onlive passport is registered as 'collector' of the card and the owner of card will be able to see who this collector is.
- If you publish a card from somebody else, your Onlive passport is registered as a 'publisher' of the card and the owner of the card will be able to see which collector has published his card in what group.
- If you reuse a card from somebody else to make your own card (with the same plugin) your Onlive passport is registered as 'Owner' of the card.

The idea is that a card always has a back side (you can flip a card on it's back) and there you can see where the card originates from, how many times it has been collected, republished and of course, who is the owner.

## User stories / scenarios

1. Register new profile (M)
    a. Name / surname (M)
    b. Email address (M)
    c. Photo (s)
    d. Phone number (W)
    e. Using Google / Facebook (S)

        2. Register organization axis (S)
3. Register as host (C)
4. Change account details (S)
    a. Personal data (S)
    b. Group permissions (C)
5. Switch accounts (C)
6. Log-in / log-out (S)
7. Password / username recovery (S)

## Epics MVP

### Register new profile (M)

User registers a new profile within the onlive platform with a name and email address

# 9. Notes

## Module goal

Connecting personal notes to cards

## UX framework (meaning)

The user can stick notes to cards. There can be a personal and a public note
Personal notes are only for the user but visible public notes can be published back to the stage.

## User stories / scenarios

1. Create new note (S)
2. Save note card to (S)
3. Choose public or private (S)

## Epics MVP

-