a

SOLID Apps

Solid Beer app

Functional Design Specification D0001SBA NL

Edition Control Chart				
Edition	**Date**	**Status**	**Changes**	**Affected pages**
0.3	2-9-2019	Draft		
0.4	18-9-2019	Draft		

Document Authorization			
	Prepared	**Checked**	**Approved**
Name/Signature	Pieter van Everdingen		
Date	18 September 2019		
Name/Signature			
Date			
Name/Signature			
Date			
Name/Signature			
Date			
Name/Signature			
Date			

Document information

Document

Document code	: D0001SBA
Document group	: Development Documents
Document title	: FSD Solid Beer app
Edition	: Edition 1.0
Month modified	: September 2019
Product Group	: Solid Apps
Location	: Utrecht/Apeldoorn, NL

Table of contents

Management summary

The ambition of Solid (Social Linked Data) is to give persons and organizations the control back over their personal data via Personal Operational Data Stores (PODS) and interoperable Solid apps. This means that persons and organizations should be able to securely manage, maintain and share their personal data themselves within the open and transparent Solid ecosystem.

This Solid Beer app experimentation activity is a collaboration between community members of Platform Linked Data Netherlands (PLDN) and the Cadastre.

The *goal* of this Solid experimentation activity is to get answers for the following questions:

For companies like the Cadastre:

- How can Solid work within a business environment?
- Which Solid apps can help to improve the service levels of the Cadastre?

For Solid app developers:

- How to develop interoperable apps with open internet standards like HTML, CSS and JS, a minimal set of additional Solid libraries and reusable (linked) data shapes?

For Solid POD and app users (persons and organizations):

- How to manage, maintain and share your personal data within PODS in a user friendly and secure manner with interoperable Solid apps that also facilitate social interaction with different persons and organizations in social networks?

For Solid POD providers:

- How to become an official POD provider in the Netherlands (next to e.g. the Inrupt and Solid Community POD providers and experimental POD providers)?

We will use a Solid Beer app use case to give answers to these questions. We already have brewery and beer linked data available in a dBeerPedia dataset and a Solid Node server in our Triply experimentation environment (Cadastre and PLDN environment). And we know from previous Linked Beer activities that it is relatively easy to mobilize Linked Data experts from PLDN and the Cadastre to participate in these activities. The team members for this Solid Beer app experimentation activity are: Erwin Folmer, Pano Maria, Dimitri van Hees, Joost Farla, Özcan Seker, Gerard Persoon, Wouter Beek and Pieter van Everdingen.

0. Revision Information

This Functional Design concerns a new Solid app and, as a consequence, revision information is not applicable.

1. Introduction

1.1 Purpose

The Solid Beer app is a mobile app that can be used by brewers and beer consumers to store and share their brewery and beer data.

1.2 Scope

The initial release of the Solid Beer app will only contain the data of the Dutch breweries that are available within the brewery and beer linked dataset of the Stichting Erfgoed Nederlandse Biercultuur (https://www.nederlandsebiercultuur.nl/). A linked dataset of this data is available at http://data.labs.kadaster.nl/dbeerpedia/dbeerpedia/, but we might decide to move this dataset to the PLDN experimentation environment in the near future: https://data.pldn.nl/.

The Solid Beer app can be a Progressive Web App (PWA), where we develop one code base for multiple platforms (Android, iOS, etc.) and where the web app runs in a mobile web browser. But for a first prototype, it can also be an Android app. Another benefit of a Progressive Web App that only need a URL to make the app available for the users. So, you don't need to work via an app store to make an app available. It is just a (mobile) web app.

1.3 Definitions, acronyms, and abbreviations

This paragraph gives an overview of the terms, aconyms and abbreviations that are in the scope of our Solid Beer app experimentation activities. In this first version we refer to the English WikiPedia for more information of items in the lists, but where more appropriate we will refer to the official W3C and/or Solid resources in a future version of this paragraph. This lists also show technology options we will discuss in the near future to decide on the set of technologies that we will use for the initial release of the Solid Beer app.

Definitions

Access Control List	
App data	
App engine	
App GUI	
App market	
Application Programming Interface	
Beer consumer POD	
Beer menu	
Beer venue	
Bootstrap	
Brewery POD	
Browser limitations	E.g. when you create a PWA with service workers?
Cache	
Camera access	
Capacitor	(formerly Apache Cordova?)
Central search index for public data	
Client server	
Client side shape validation	
Closed group	
Communica	Meta query engine for the web to execute SPARQL queries on files that are downloaded from PODS
Compatible data shapes	
Context	
Cross-platform development	
Data graph	
Data grid	Or knowledge grid (within the context of gossiping)
Data market	
Data model	
Data ownership	
Data worker	
dBeerPedia	dBeerPedia linked dataset and ontology, created for development and demo purposes, which contains the descriptions (semantics/meaning), definitions (metata), relationships (data model structure) and instances of the brewery and beer data of Dutch breweries
Decentralization	
Distributed data	
Domain Specific Language	
Elasticsearch	
File system	
Footprint	
Form	
Full text search	

Geo centric query	
Google mapping	
Gossiping	
GraphQL-LD	
Group Administrator	An app user that initiates group activities by creating a group and sending invitations to group members
Group Member	
Hybrid app	
Identity provider	
Interoperability	
Ionic	
Interoperable apps	
JavaScript	
JQuery	
LDFlex	
Linked Data	Structured data in an RDF-format, which is accessible and connectable on the internet via URI's in a uniform way
Linked Data Notifications	
Linked Data Platform	
Location aware app	
Location Based Social Network	
Mashlib.js	
MEAN stack	
Message queue	
Metadata	
Mobile app	
MongoDB	
N-Quad	
N-Triple	
Native app	
Notfication/inbox mechanism	Mechanism to send and receive messages when using an app
Notifications	Messages that are send to friends (directly and when they are a member of a group)
Ontologies	
OpenAPI's	
Peer-to-peer search	
Personal Operation Data Store	
POD provider	
Portability	
Privacy by design	
Private data	
Private folder	

Progressive Web App	
Public data	
Public folder	
Public keys	
Publish-Subscribe	
RDF data model	
RDFlib.js	
Resource Description Framework	
Reusability	
Reusable RDF data models	Data models in RDF for different apps in a certain domain, where the data models of these apps are based on the same vocabulary / vocabularies and have a similar structure (shape)
Router	
Scalability	
Server side shape validation	
Service worker	
Shape	SHACL modelling construct which defines a validation constraint on the allowed structure or on the allowable values of data instances in a knowledge graph
Shape graph	
Shape validation	Data instances validations process where data instances are checked to decide whether these instances comply to the shapes that apply to these instances
Schema.org	
Semantic wrapper	JavaScript script(s) that makes it possible to work with linked data (RDF) in an app
Social Linked Data	
Social Semantic Web	
Solid identity	
Solid POD	
Solid violation	
solid-auth-client	Browser library that allows your apps to securely log in to Solid PODS and read and write data from them
solid forms	
solid-panes	
SPARQL UPDATE query	SPARQL query that updates data in a POD or a collection of connected PODS
State management	App state management
Triple	
Triple store	
Turtle	
User location	
Validation report	
Vocabularies	
Vue.js	

Vue Native	
Vuex	
Web app	
WebID	
WebSockets	

Acronyms

ACL	Access Control List
AVG	Algemene Verordening Gegevensbescherming
API	Application Programming Interface
CLI	Command Line Interface
CRUD	Create, Read, Update, Delete
CSS	Cascading Style Sheets
DSL	Domain Specific Language
FOAF	Friend Of A Friend vocabulary/ontology
GPS	Global Positioning System
HTML	Hypertext Markup Language
HTTPS	Hypertext Transfer Protocol Secure
IP	Internet Protocol
IRMA	I Reveal My Attributes, a Dutch online identity platform that support privacy-friendly authentication and digital signatures processes
JS	JavaScript
JSON-LD	JavaScript Object Notation-Linked Data
LBSN	Location Based Social Network
LDP	Linked Data Platform
MEAN	MongoDB, Express.js, AngularJS (or Angular), and Node.js stack
MERN	MongoDB, Express.js, React, Redux and Node.js stack
OIDC	OpenID Connect
NPM	Node Package Manager
PODS	Personal Operational Data Store
pubsub	Publish Subscribe
PWA	Progressive Web App
RDF	Resource Description Framework
SDK	Software Development Kit
SHACL	Shapes Constraint Language
SOLID	Social Linked Data
SPA	Single Page Application
TCP	Transmission Control Protocol
TLS	Transport Layer Security?
WAC	Web Access Control
WAC-LDP	Web Access Control and Linked Data Platform
WebID-OIDC	WebID and OpenID Connect
WSS	Web Services Security

1.4 References

Using the links of the following references will navigate you to some of the official Solid resources on the internet with additional information and guidelines about Solid:

Tim Berners-Lee: The next Web of Open, Linked Data (TED Talk in 2009):

- https://youtu.be/OM6XIICm_qo

Official Solid Resources:

- https://solid.inrupt.com/
- https://forum.solidproject.org/
- https://gitter.im/solid
- https://github.com/solid
- https://github.com/solid/information

Solid presentation by Ruben Verborgh (video + slides):

- https://ftp.heanet.ie/mirrors/fosdem-video/2019/Janson/solid_web_decentralization.mp4
- https://rubenverborgh.github.io/Slides-FOSDEM-2019/#

Blogs by Ruben Verborgh:

- https://ruben.verborgh.org/blog/2019/06/17/shaping-linked-data-apps/
- https://ruben.verborgh.org/blog/2018/12/28/designing-a-linked-data-developer-experience/

Solid events in the Netherlands:

- http://www.pilod.nl/wiki/PLDN-Solid_Kick-Off_%E2%80%93_April_9th_2019
- http://www.pilod.nl/wiki/Solid_The_Hague_Kick-Off

Make a Solid app on your lunch break (JQuery & React version):

- https://solid.inrupt.com/docs/app-on-your-lunch-break
- https://github.com/solid/profile-viewer-react

Prize winning Solid chat app source code example and documentation:

- https://github.com/Arquisoft/dechat_en2a
- https://arquisoft.github.io/dechat_en2a/docs/

Solid Chess app, source code, documentation and articles:

- https://pheyvaer.github.io/solid-chess/ (chess app)
- https://github.com/pheyvaer/solid-chess (source code)
- https://pheyvaer.github.io/solid-chess/docs/index.html (documentation)
- https://pieterheyvaert.com/blog/2019/02/10/solid-world-summary/ (intro)
- https://dev.to/phadventure/how-a-chess-app-interacts-with-solid-409a (text from previous resources combined in one article with sequence diagrams for a number of processes plus RDF data models and SPARQL query examples)

1.5 Overview

This document contains the following elements:

- A description of the overall requirements of the Solid Beer app;
- A discussion of the components involved;
- A description of the user interface of the Solid Beer app.

2. Overall requirements

2.1 Environment

The Solid Beer app is designed to run on mobile phones. The initial release will be written using open standards like HTML, CSS, JavaScript, RDF-based standards like Turtle, possibly in combination with libraries/components from Ionic and Vue.js

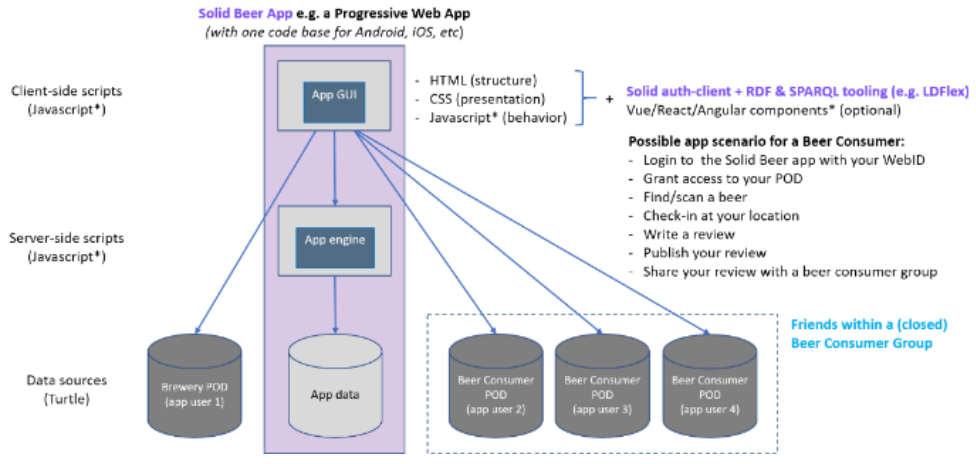
Untappd is developed with *Ionic*, which makes it possible to develop *one code base* for an app for multiple platforms (Android, iOS, etc.). The web app runs in a mobile web browser then (a wrapper). This adds an extra layer to the app, which will cause a small performance penalty when brewers and beer consumers are using the app. An additional benefit of using Ionic is the speed of development given the fact that Ionic comes with 100+ libraries with components to develop cross platform mobile apps. Ionic can be used with Angular and other JavaScript development libraries/frameworks, but React and Vue.js are in Beta now.

Preferably we would like to use the Enterprise version of Ionic for free until March 2020, since this is a PLDN community activity with a small budget.

2.2 Architecture

As mentioned above the Solid Beer app is a mobile app that brewers and beer consumers can use to store and share their data. It provides mechanisms for beer consumers to create profiles, groups, write and share reviews, etc. and for brewers it provides mechanisms to share their brewery and beer data with others in a uniform way via standard OpenAPI's.

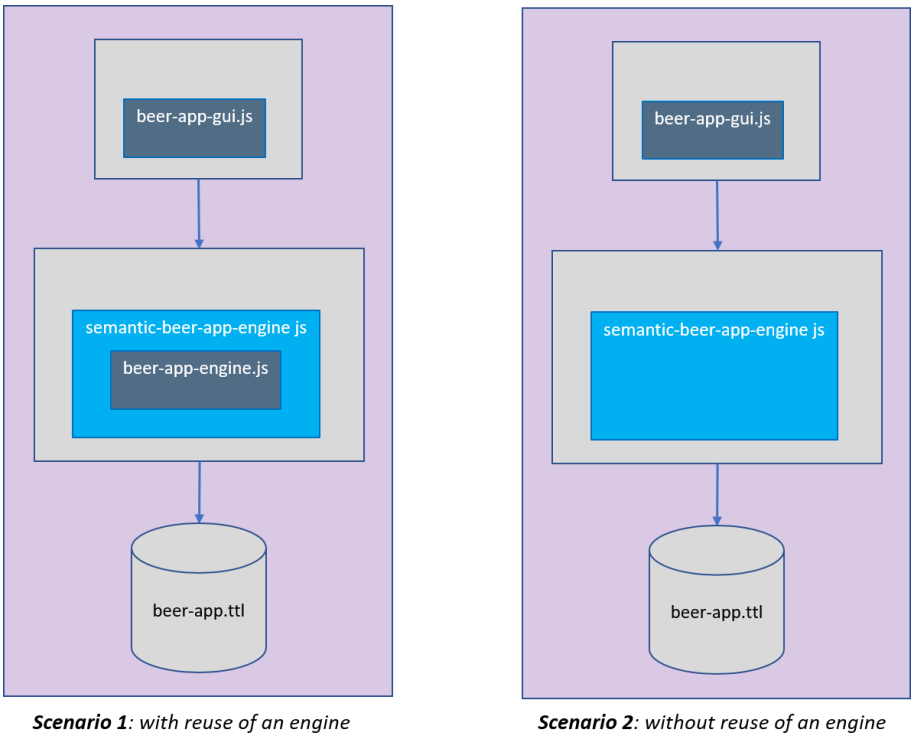
The following diagram gives an overview of the Solid Beer app components:



Figuur 1 – Solid Beer app (high level) application architecture

The main components of the app are:
- Solid Beer app GUI
- Solid Beer app engine
- Solid Beer app data
- Brewery PODS
- Beer Consumer PODS

And when we zoom into the app itself then we see two possible scenarios:
- Scenario 1: with reuse of an existing engine
- Scenario 2: without reuse of an engine



***Scenario 1**: with reuse of an engine*

***Scenario 2**: without reuse of an engine*

Figuur 2 – Possible Solid Beer app application architecture scenarios

Scenario 1 has been used during the development of the Solid Chess app, since a chess.js engine was available for reuse. But this chess engine did not work with Linked Data and needed an additional semantic-chess.js wrapper for this reason that it can work with the available Linked Data in PODS and of the app itself. But for development of the Solid Beer app we will follow scenario 2, since no reusable engine is available.

The names of the different Solid Beer app components are work titles to give an idea what the role of the component is in the Solid Beer app application architecture. Our lead developer Ozcan Seeker will decide what the final naming of the components will be with the support of the PLDN/Cadastre Solid Beer team.

2.3 User characteristics

The Solid Beer app should be an intuitive and easy to use mobile app. This means that the brewer and the beer consumer don't need to have any additional knowledge or skills to make use of the Solid Beer app. One exception might be when a brewer wants to upload new brewery and/or beer data to its own POD and to the Solid Beer app. Depending on how this implemented this might ask for some 'special' skills, but the general aim of the Solid Beer app and the PODS is to make them as easy as possible to use.

2.4 Constraints

2.4.1 Performance Requirements

Performance depends on many factors, like type and capacity of a mobile phone, its current usage pattern by the user, available memory and storage, bandwidth, etc. And it depends on the Solid Beer app configuration. In general, we can say that a user would like that an activity on their mobile phone takes less than a second. But for the initial release of the Solid Beer app and for demo purposes we can state that an app activity should take less than 1 second in general and 5 seconds for a limited number of (heavy) activities, but not longer.

2.4.2 Software System Attributes

- **Interoperability**: The Solid Beer app must use compatible data shapes for the personal data that is stored in the Beer Consumer PODS in such a way that other Solid Beer apps can recognize these shapes and can make use of these shapes that no personal data gets lost when a beer consumer switches from one Solid Beer app to another.
- **Portability**: The Solid Beer app must be able to run on different mobile platforms (Android, iOS, etc.), but at minimum on Android for the initial release.
- **Security**: The security is based upon the standard Solid security mechanisms that makes use of public keys, WebID's in the auth-client and Access Control Lists.
- **Availability**: The Solid Beer app must be available 24/7. But given the fact that we make use of a PLDN experimentation environment this requirement is less strict.
- **Reliability**: The initial release of the Solid Beer app must be sufficiently stable, that it can be used for live demo purposes.
- **Scalability**: The Untappd app processes millions of check-in events worldwide per month and makes use of different databases (MySQL, MongoDB, Redis and Elasticsearch) in combination with a database as a service management system (ObjectRocket), a message queue and router (IronMQ and IronWorker from Iron.io), to create a highly scalable Untappd environment. For the initial release of the Solid Beer app we expect to have a limited number of events that we will create for experimentation purposes and we don't have the intention to compete with Untappd, we just want to show the possibilities of Solid and Linked Data with an interesting and easy to understand use case.
- **Maintainability**: The code base of the Solid Beer app should be as maintainable a possible (preferably one code base, correct usage of layers, separations of concerns, modular setup, decoupled, standard OpenAPI's, sufficient documentation, etc.).

Interoperability, portability and security are the high priority software system attributes for this Solid Beer app experiment. And it would be nice if can spend some time of the other software system attributes when time and budget allows us to do so.

FAIR principles nog toevoegen

2.4.3 Other Requirements

It would be nice if this Solid Beer app experiment also leads to a first Solid best practice for developing interoperable apps in the Netherlands (see also next paragraph).

2.5 Horizontal Issues

Solid is a global initiative and it would be nice if our activities also contribute to the global Solid ecosystem, platform and community. This is also one of the reasons to create ‘official’ Solid Beer app documentation in English that no translation is needed when we would like to make use of the global Solid expertise in the Solid community and involve Solid experts worldwide (when needed and when appropriate). And we can switch back to Dutch during our local discussions and record decisions and other documentation in English.

We start locally with a small team in the Netherlands. Our experience is that this works more efficient and leads faster to results. But where needed we would like to involve other experts.

3. Data models

The Solid Beer app contains several data components, each with their own data structure (data shapes/patterns). The following sections will describe the different data components in more detail given the scope of the Solid Beer app.

3.1 App data

Starting point for the App data is an existing Linked Beer Turtle file that we have available from earlier Linked Beer activities. And we can make this Turtle file up to date with the latest information that is available on the website of Stichting Erfgoed Nederlandse Biercultuur (see: https://www.nederlandsebiercultuur.nl/). Then we must decide what additional app data we need to support the processes as described in chapter 4 and what data is app data and what data is personal data of breweries and beer consumers that will be stored in their PODS.

Creating the more detailled requirements and design for the app data will be done in a later phase when we have agreed on the scope and level of detail of the initial release of the Solid Beer app. Most-likely we will make this more concrete using a prototyping approach, where we use, validate and improve the process descriptions of chapter 4 and the screen sketches of chapter 5. This design document is just a start document for our Solid activities.

3.2 PODS data

Starting point for both PODS is a default Solid POD with very basic profile information. For both PODS we must decide what additional properties we need to create Brewery and Beer Consumer profiles with sufficient detail. And we must also decide how to implement these profile extensions consistently according the official Solid guidelines. The following Solid documentation gives some pointers on how to implement an extension:

https://github.com/solid/solid-spec/blob/master/solid-webid-profiles.md

The following sub paragraphs give a first idea of what additional profile data we would like to store in the Brewery PODS and the Beer Consumer PODS.

3.2.1 Brewery PODS data

First thought on this, is to state that a brewery is just a special kind of organization and that we can use the property descriptions of an existing often used organization vocabulary to add the required additional properties. But we have these properties already available in the existing Linked Beer Turtle file. But we must decide then how to shape this data for storage in the Brewery POD that it can be shared and re-used by others using standard OpenAPI's.

The brewery information is 'just' a kind of profile data, but the beer data will most-likely be stored in a beer.ttl file in the public folder of the Brewery POD, where we would like to have a compatible beer shape that is recognizable and usable for others to re-use in their apps or websites. The Brewery POD is THE authorative source for the brewery and beer information of that brewery, which can be used (linked) without unnessarily copying that data.

See appendix … for an overview of the brewery and beer classes, properties and relationships (this appendix will be added onces we have the Linked Beer Turtle file available).

3.2.2 Beer Consumer PODS data

Similar to a brewery, we can state that a beer consumer is just a special kind of person and that we can use the property descriptions of of an existing often used person vocabulary. And when we need additional properties that are not available in any often used vocabulary, we can define those properties ourselves (if needed). We must also decide how to store our interactions with the Solid Beer app. Will it be one 'big' beer-check-in.ttl or is it advisable to have a number of smaller, overlapping Turtle files, where re-usable shapes are more easy to recognize.

In the beer-check-in.ttl you collect all personal data you want to store in your POD after you have checked in at a certain location. All interaction data is then just in one file. But we can also decide that for the initial release of the Solid Beer app we just have a beer-reviews.ttl file in the public folder of our POD that can be easily re-used by other beer apps (if any).

See appendix … for an overview of the beer consumer class, properties and relationships (this appendix will be added during the prototyping phase of our Solid activities).

3.3 Other (linked) data sources

The existing Linked Beer Turtle file might also contain information about *beer venues* (e.g. the pubs in the Netherlands) and we see in Untappd that a limited number of pubs also publish their *beer menu's*. This data is out of scope for the initial release of the Solid Beer app, but when we have time available we can brainstorm on how we can implement this data in the Solid Beer app and how brewery and beer data from a Brewery POD can be re-used in the web site of a pub using standard OpenAPI's and without copy-ing that data.

The beer community is typically a *location-based social network* (LBSN). This means that it also makes sense to make use of the GPS-functionality of a mobile phone and to make use of geo data and maps to add functionality for the user that he can find beer-related information of a certain location or in the neighborhood of a certain location. Also this is out of scope for the initial release of the Solid Beer app, but also for this feature we can think of how we can implement geo-centric queries and functionality when we have time available.

For us, it would be nice if we can have, next to the sources that we already have, at least one additional simple Linked Data source that we use in the total Solid Beer app configuration that we can highlight the strength and the flexibility of Linked Data. Maybe it is easy to triplify an existing beer festival calendar in the Solid Beer app. But, we can also decide in a later phase on another source if we can think of something better than a beer festival calendar.

4. Process model

The Business Component Browser does not implement a business process, per se. Instead, it is a technology component that roughly implements an extended Elementary Process. As a result, the prescribed diagrams are inappropriate.

In lieu of process dependency and process decomposition diagrams, class diagrams that describe the structure of the browser are provided below.

4.1 Brewery processes

4.1.1 Create a Brewery POD with a brewery profile

A brewery company gets a POD via the default Solid mechanisms to get a POD and a WebID:

- https://solid.inrupt.com/get-a-solid-pod

At this page you can choose between two POD providers to host your POD: 1) Inrupt and 2) The Solid Community. The difference between the two that the Inrupt only has a public folder to store your data and the Solid Community POD has a public and a private folder to make the distinction between the data that your would like to share with other persons and organizations and the data that you would like to keep private by default (your 'internal' data).

PLDN has installed a Solid Node server in their Triply experimentation environment, which can also be used to get your own POD:

- https://solid.pldn.nl/

This PLDN Solid Node server is similar to the Solid Community Node server, where PODS also have a private folder. In practice it should make no (major) difference, which POD provider you choose, but within the Solid community there is a discussion going on whether a private folder is needed in a POD or whether everything can be done with just one public folder. Based upon this information, we can make a decision which provider(s) to use.

For e.g. the brewery Brouwtoren, possible WebID's of a Brouwtoren POD can be:

- https://brouwtoren.inrupt.net/profile/card#me
- https://brouwtoren.solid.community/profile/card#me
- https://brouwtoren.solid.pldn.nl/ or
- https://www.brouwtoren.nl/profile/card#me

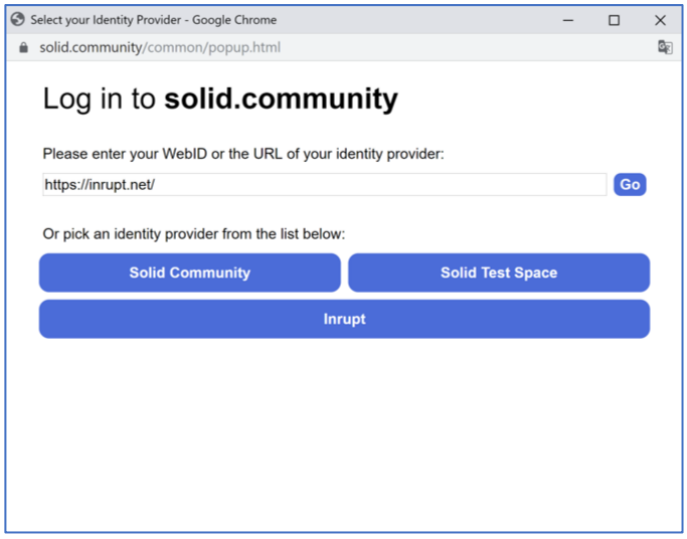
The Solid Node server in the Triply experimentation environment still has a security certificate issue, which will cause a security warning when you try to access a POD. We hope that this will be solved soon.

4.1.2 Authenticate to the Solid Beer app with your WebID

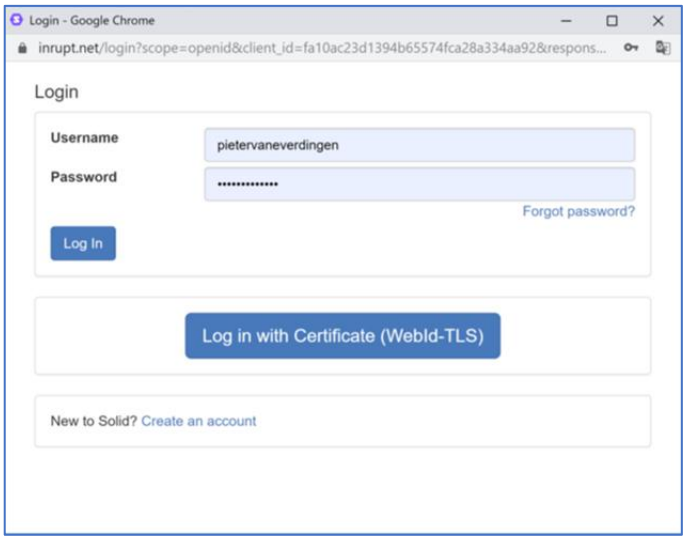
A Solid app must give you the possible authentication options for the PODS of the POD providers they support. Usually this is a screen with the provider options:

- Solid Community
- Solid Test Space (is no longer available)
- Inrupt
- Solid PLDN (we can add a button like this for our Solid environment)
- …

In most cases the first authentication screen looks something like this:



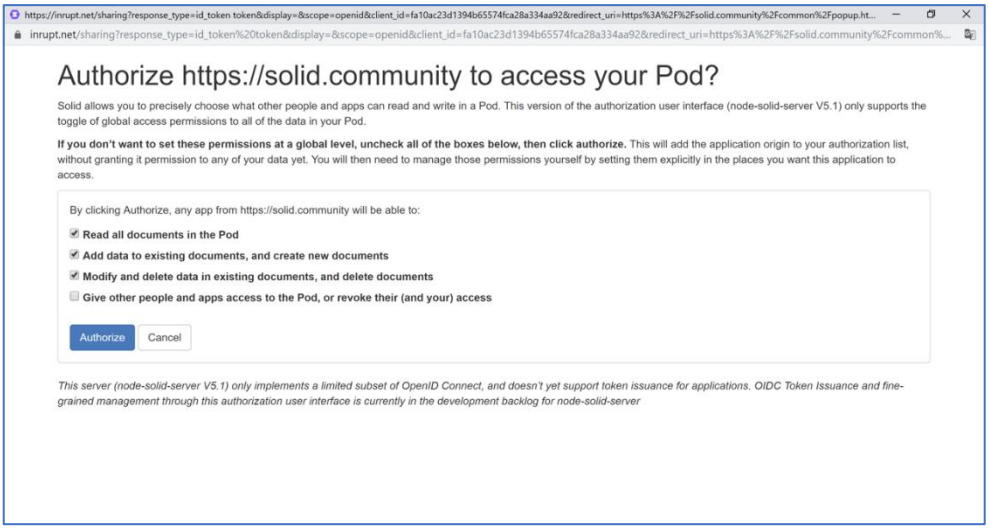
And after you have chosen your provider you can login to the app. And a ‘default’ Solid login screen looks like this:



A few apps have a ‘custom’ login screen that is styled with the same look and feel as the rest of an app. For the initial release of the Solid Beer app, this has no high priority.

4.1.3 Grant access to the Solid Beer app to use your Brewery POD

The last step during the login process is that you have to grant access to the app that it can make use of the data in your POD and can write new data in your POD. A grant access screen usually looks something like this:



4.1.4 View the brewery profile home page

The brewery profile screen is the opening screen for a brewer when he opens the Solid Beer app. It shows the brewery information and the average score of that brewery, a list of the beers that are brewed by the brewery, also with the average score of each beer and the location of the brewery. Usually this information does not fit on one screen, which means that you have to scroll down to see all information. And for certain screen sections it can also show only a subset of the data for that section with a Show all button of the bottom of that section.

For the initial release of the Solid Beer app we must decide how much detail we would like to add for a good user experience. It would be nice if we can click on beers in the list of beers of all breweries to navigate to the beer description screens from a list. Untappd has much more information linked to their brewery profile screens. We can have a look at that if we have time available, but it has no priority to include this information in the initial release of the Solid Beer app (and maybe it is even too much information that is not often used there).

4.1.5 Synchronize brewery and beer data

This does not sound like a mobile app feature, but more like a feature that you would like to execute on your laptop. So, for the brewer there might be a few Solid Beer app web pages that he can use for administrator tasks. For the Solid Beer app we assume that the developer has used an existing Linked Beer Turtle file to load an initial set of brewery and beer data into the Solid Beer app. But it must also be easy for a new brewer to start from scratch and add their brewery and beer to the Solid Beer app via an easy to use upload API.

4.1.6 Maintain brewery and beer data

Also this process sounds more like a feature that you would like to execute on your laptop. Also this is more an administrator task, where your would like to make use of an easy to use upload API to add new beer data or change existing brewery or beer data. One additional comment that we would like to make on this topic is that we assume that the brewery and beer data is stored in a Brewery POD. So, this process must be able to add new beer data in a POD or change existing brewery and beer data in a POD and upload a new version of the brewery and beer data to the Solid Beer app that the data in the app is always up-to-date.

4.1.7 Share brewery and beer data

Next to the Solid Beer app, brewers might also want to share their data with other persons and organizations, like beer venues. Beer venues, like pubs can use this data then to publish their beer menu with the right information from the Brewery POD. Via ‘Beer API’s’ a venue must be able to get the beer data from the beers that they have on their beer menu. To make this happen we can ask a pub in our network to participate in our Solid Beer activities that we can get a more founded idea of how this could work and how easy or difficult it is to implement this. It sounds ‘do-able’ for a proof concept to get a better idea for this.

4.2 Beer Consumer processes

4.2.1 Create a Beer Consumer POD with a consumer profile

This process works exactly the same as the creation of a Brewery POD. See paragraph 4.1.1 for a more detailed description of this process.

4.2.2 Authenticate to the Solid Beer app with your WebID

Also this process works exactly the same as the authentication process for a Brewery POD. See paragraph 4.1.2 for a more detailed description of this process.

4.2.3 Grant access to the Solid Beer app to use your Beer Consumer POD

And also this process works exactly the same as the grant access process for a Brewery POD. See paragraph 4.1.3 for a more detailed description of this process.

4.2.4 View the beer consumer profile home page

The beer consumer profile screen is the opening screen for a brewer when he opens the Solid Beer app. It shows a profile picture, the number of check-ins, the number of beers reviewed, the user name, the usage start date, the beer bonus points, a beers link, a friends link, a groups link (not in Untappd in the profile screen) and a recent activities list. Also for this process the shown information does not fit on one screen, which means that you have to scroll down to see all information. And for certain screen sections it can also show only a link to more detailed information, like for beers, friends and groups. We can decide to make only a small part of this data public that a profile can be found and that the more detailed information is only visible for your friends and friends within beer consumer groups (a more closed approach).

4.2.5 Scan the barcode of a beer

The most easy way to find a beer description is to scan the barcode on a bottle or can of beer.

The profile (home) screen has a search/scan field at the top of the screen (similar as Untappd) that opens an (existing) barcode scanner ones you click on the scan section of that field. Put the barcode in focus on the screen and you hear a beep after the barcode has been recognized. You will navigate then to the matching beer description screen of that bottle or can.

4.2.6 Search for a beer via the search field

Another way to find a beer, e.g. when you have a glass of beer in a pub, is by using the search field in the profile (home) screen. In Untappd you have several options to find a beer:

- Indirectly via the brewery name
- Directly via the beer name
- Indirectly via a location

For the initial release of the Solid Beer app we will only support the second search option.

Ones you type something in the search field, the rest of the screen will change into a search results screen. This results screen will show the possible matches with the search entry in a list. You will navigate to the matching beer description screen after you have clicked on the correct beer option in the results list. When feasible within the available time, we can add the other search options (Utappd alike), but this has no high priority.

4.2.7 Check-in at a location

Onces you have found the beer that you are drinking now and that you want to review, you can click on the check-in button to check-in at a location and to open the beer review screen (see next paragraph). For this activity in the app you will receive 5 beer bonus points that will be added to your beer bonus points total. See paragraph 4.2.9. for more details on the review and share award system.

4.2.8 Write and post a review for a beer

So, you must check-in to be able to write a review (see previous section). Therefore, use the check-in button in the beer description screen to open the beer review screen.

Use the slider in this review screen to enter the rating for this beer (mandatory) and add a short comment with your beer drinking experience details and how you appreciate or dislike the beer (optional). This can be a free text field of e.g. 280 characters. You can also add a picture of the beer that you are drinking at this location that you have taken with the camera of your mobile phone (optional).

For the initial release of the Solid Beer app we would like to have the rating and the comment feature in the app. Adding a picture to a review has low priority.

Onces you are ready writing the beer review, you can click on the post button (Publiceer button) to add the review to the recent activities of your profile (home) screen. For this activity in the app you will receive 10 beer bonus points that will be added to your beer bonus points total. See paragraph 4.2.9. for more details on the review and share award system.

4.2.9 Connect to friends

Cllick on the Friends (Vrienden) link in the profile (home) screen to go to the friends screen. This screen shows a list of friends the beer consumer has already connected to one ore more friends or is empty for a new Beer Consumer user. This screen also contains a search field at the top of the screen. You can type a name into this field and see possible friends that match with this search entry in the search results screen. When you see the name of your friend in this list, then you can click on his/her name to go to the profile of this friend.

This profile screen will now a Send friendship request (Vriendschapverzoek sturen) button. Click on this button to send a friendship request to this friend. This friend receives this invitation in the message box of the Solid Beer app and he/she can confirm this friendship by clicking on the Accept friendship request (Vrienschapsverzoek accepteren) button. After the confirmation the new friend will be shown in the Friends list. And after a beer consumer has one or more friends he can create a closed beer consumer group.

4.2.10 Create a closed beer consumer group

The major difference between Untappd and the Solid Beer app is that we would like to have our beer experiences data a bit more private. To do this we make this process more visible in the user interface e.g. via a direct link to Groups (Groepen) in the profile (home) screen of a Beer Consumer. After you click on this Groups link you navigate to the Groups screen, where a list of groups is shown if a beer consumer has already created one ore more groups or is empty for a new Beer Consumer user.

This screen also contains a Create a new groep (Nieuwe groep aanmaken) button. Click on this button to create a new group in the Create a group screen. Enter a name for this group in the Group name field and click on the create group now (Groep nu aanmaken) button. The next screen will show the friends for your friend list. You can select now the friends from your friends list that are also shown in the Create a group screen. Click on the Send invitation to join (Uitnodiging naar vrienden sturen) button in this screen.

The selected friend(s) receive the invitation to join the group in their message box of the Solid Beer app and they can confirm this friendship by clicking on the Accept grpup partcipation request (Groepsdeelnameverzoek accepteren) button. After the confirmation the new group will be shown in the Groups list of each friend that accepts the invitation and in the Groups List of the Beer Consumer that has created the beer consumer group.

4.2.11 Share a review within a beer consumer group

After you have written a beer review, you must have the opportunity to share this review with your friends within a beer consumer group via a share button (Deel button) in the beer review screen. You can open the beer review by clicking on it on you profile (home) screen. Then you see the details of the review that you have written. This screen shows also a share button that you can use to share this beer review with your friends in a beer consumer group. After you click on the share button you see list with your beer consumer groups (one or more). Select the group(s) you want to share the review with and click on the share now button. Each member of the group(s) you have selected will receive a message now with your review.

For this activity in the app you will receive 25 beer bonus points per group that you share the review with, that will be added to your beer bonus points total. See paragraph 4.2.9. for more details on the review and share award system.

4.2.12 Receive beer bonus points for check-in, review and share activities

One of the reasons to start the Solid Beer app development activities was that we wanted to start a new competition from scratch with the persons involved.

The award system in the Solid Beer app is simple:

- You receive 5 points when you check-in to a location
- You receive 10 points when you write and post a review
- You receive 25 points when you share your review within a beer consumer group

This means that the app must store the total bonus points score until now and add the new points to this total score for every new app activity as described above. It also means that you receive most beer bonus points for being social with your beer friends.

Your total score is shown in your profile screen in the beer bonus points section.

4.2.13 Show beer bonus points ranking in a beer consumer group

In the beer consumer group we can by default sort the members of the beer consumer group according to their ranking within the group. We don’t need a separate screen then to show the ranking within a group.

5. User Interface

5.1 User Roles

User Role BREWER

Definition: A person that works for a brewery and brews one (or more) specific beers within that context and would like to share his brewery and beer data with others.

Description: Brewers will use the brewer administrator screens to upload and maintain their brewery and beer data and share their data with others in a uniform way.

User Role BEER CONSUMER

Definition: A person with an interest in beer that can become a member of a closed beer consumer group and that would like to share his/her beer expertise and consumption experiences with other beer consumers via reviews.

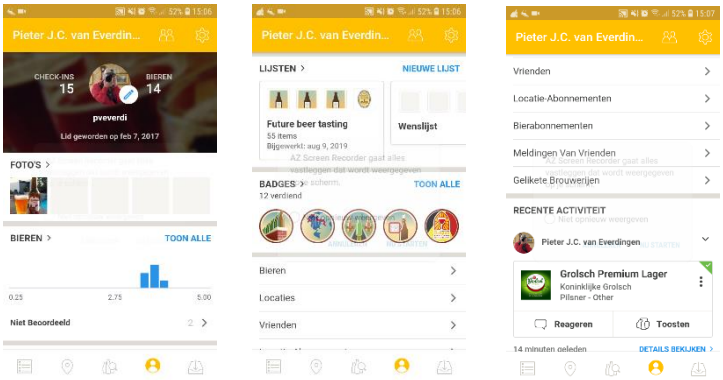
Description: Beer consumers use the screens within the Solid Beer app to create beer consumer profile, a beer consumers group and to write and share reviews with other members of a beer consumer group.

5.2 Screen Sketches

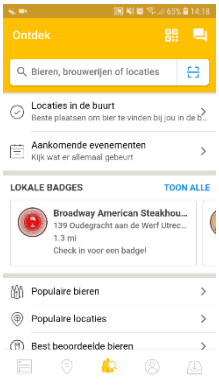
For our screen design we have looked at the screen design of the Untappd screens to get an idea of the screen lay-out and GUI elements we would like to add on the screens of the Solid Beer app. So, the overall design might look very similar to Untappd, but we are using the PLDN colors white, dark blue and medium blue, with white and dark blue text to make the look and feel a bit different from Untappd. And also given the more limited scope of our Solid Beer app we might also show less detail in some of the Solid Beer app screens.

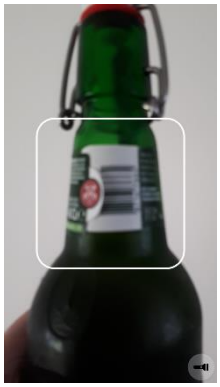
The following screen examples are added to give a first idea of how the screens might look like in our Solid Beer app:

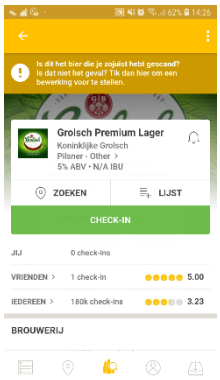
Beer Consumer profile screens (with scroll down details):



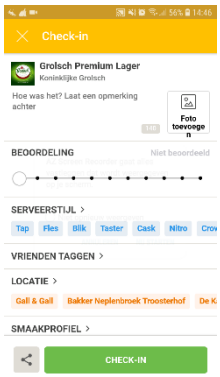
Find a beer screens

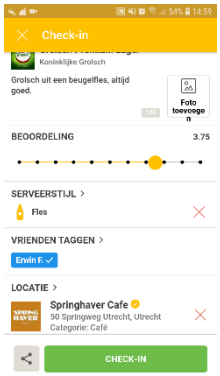






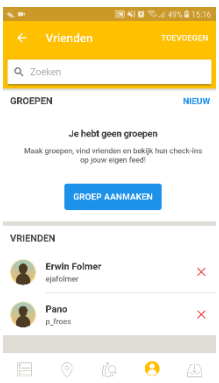
Check-in and write a review screens

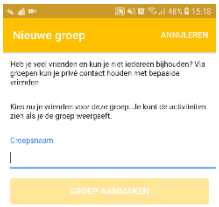




Beer Bonus Points Screens:

Beer Consumer groups screens:





Ones we have agreed upon the screens that we really need for our initial release of the Solid Beer app, we can make our UI/UX design and a screen map outline the possible navigation paths in the Solid Beer app.

5.3 Screen Map

See previous paragraph. This is more a detailed description for an appendix?

5.4 User Interface Description

As described in the previous paragraph, we will look at the Untappd screens to get an idea of how our Solid Beer app screens can look like (with less detail). And the GUI elements are also determined by the libraries we have available in our Solid development environment. Untapped has been developed using the Ionic ecosystem/framework as described earlier in this document and we can decide to use Ionic or a similar development environment with sufficient libraries and GUI elements to speed up our Solid development activities.

This paragraph will we work out in more detail ones we have chosen our Solid development environment.

5.4.1 General Browser Features

In this paragraph we will describe the most important GUI elements that will be used in the Solid Beer app to facilitate easy and consistent navigation within the app.

Tabs

Untappd works with tabs at the bottom of the home page and at the top of some of the pages for certain more detailed topics when you navigate to that topic (e.g. after clicking on the Activities, Location or Notifications tab at the bottom of the Untappd home screen.

For the initial release of our Solid Beer app, we must decide if we want to make use of tabs in several screens and if so which tabs we would like to have for each user role. The following overview gives a list of possible home screen tabs for a brewer and a beer consumer.

Possible Brewery user home screen tabs:

- Profile tab
- Beers tab?
- Venues tab? (the venues that are connected to a brewery)
- Menus tab? (the beers that are on the beer menu of the connected venues)

Possible Beer Consumer user home screen tabs:

- Profile tab
- Friends tab
- Groups tab
- Activities tab
- Notifications tab

For each tab we must decide whether we really need this tab (only if it adds to easy navigation) and where it will be positioned on a screen.

For e.g. a Brewery user, a Profile screen might be sufficient in the initial release with no tabs on the home screen. Just the brewery data and a list of beers might be good enough to keep the overview and to keep navigation simple. From the list of beers you can then navigate to the detailed beer description of each beer in the list by clicking on the name of the beer.

For a Beer Consumer user, it makes more sense to use some tabs. E.g. in the home screen we can start with a Profile and an Activities tab. Untappd doesn't have a separate Friends and Groups tab, but we can also decide to have a 'combined' Friends/Groups tab if we would like to make that screens more visible in the app and to keep the navigation very simple.

Back buttons

Show all links

…

5.4.2 Search/scan field

5.4.3 Notifications/inbox mechanism

5.4.4 Datasync API's

Appendices

In a number of appendices we will describe a few topics in more detail. The aim is to have one specification document, where the main text with the high level requirements, data, process and user interface descriptions will be stable early during our experimentation activities and that the appendices will be produced in more detail during the prototyping phase.

Appendix A – Profile extensions

In this appendix we will describe the additional properties we would like to use in Brewery and Beer Consumer profiles and how to implement those properties consistently in Brewery PODS and Beer Consumer PODS.

For the Brewery data we see the following predicates in dBeerPedia:

- <http://dbeerpedia.com/def#brouwmeesters>
- <http://dbeerpedia.com/def#categorie>
- <http://dbeerpedia.com/def#groep>
- <http://dbeerpedia.com/def#opgericht>
- <http://dbeerpedia.com/def#owners>
- <http://dbeerpedia.com/def#provincie> (*)
- <http://schema.org/address> (
- <http://schema.org/email> (*)
- <http://schema.org/name> (*)
- <http://schema.org/taxID>
- <http://schema.org/telephone> (*)
- <http://schema.org/url> (*)
- <http://dbeerpedia.com/def#bagAdresseerbaarobject>
- <http://dbeerpedia.com/def#bagNummeraanduiding>
- <http://schema.org/addressLocality> (*)
- <http://schema.org/latitude>
- <http://schema.org/longitude
- <http://schema.org/postalCode> (*)
- <http://schema.org/streetAddress> (*)

We must decide how this matches with a the data in a default Solid profile and how we want to describe the additional properties then. One scenario is to use the dBeerPedia data ‘as is’ for the experimentation activities and give a recommendation at the end of the experimentation activities on how to re-model dBeerPedia to make it as reusable as possible with the most generic and most accepted vocabularies and shapes for a real Solid app.

Our brewer Dimitri will decide what brewery and beer data he wants to store in his Brouwtoren POD and he will work with Pano to decide on what additional vocabularies we will use (if needed) to define and store additional properties in a Brewery POD. The predicates above marked with a **(*)** are already available in a Solid profile in a certain way. And we must decide how to (re)use these properties in such a way that it fits our beer use cases.

Appendix B – Compatible data shapes

This appendix describes a first detailed design for the personal data of a Beer consumer when he uses the Solid Beer app. Our goal is to define compatible data shapes that are as generic and as reusable as possible for different Solid Beer apps. For each context we will create a separate folder where simple social linked data can be stored as easy as possible.

Our first design has been based upon Pixolid and Bookmarks apps best practices. Pixolid for the modular approach with a folder structure and ttl-files for each different type of data and the bookmarks apps for having one ttl-file one type of data that can be easily reused by other apps that offer the functionality to create and store bookmarks in a POD.

Beer folder (in the root/timeline folder or in the public folder)

- **check-ins.ttl**, one ttl-file with the metadata of each check-in and where each check-in refers (links) to the beer of a check-in (e.g. via an in-reply-to predicate)
- **comments.ttl**, one ttl-file with the comments and the metadata of those comments and where each comment refers (links) to e.g. a review
- **friends.ttl**, these friends are also friends in your Solid profile (store only data at this location which is needed for this context?)
- **groups.ttl**, this can also be a group in your address book in your public folder (store only data at this location which is needed for this context?)
- **images** sub folder (Pixolid alike)
    - images (list of images, each with their unique ID)
    - metadata of the images in a ttl-file per image
- **likes.ttl**, one ttl-file with the beers you have liked and when
- **notfications.ttl,** use this in combination with your Solid inbox for messages, invitations, etc. (one uniform messaging mechanism for all communication with friends?)
- **reviews.ttl**, met de reviews die gelinkt worden aan check-ins

Brewery folder (in the root/timeline folder or in the public folder)

- **likes.ttl**, one ttl-file with the breweries you have liked and when
- **…**

We must decide how to define this data structure and the more detailed data shapes as modular and as generic as possible. E.g. if we decide to have a first version of the Solid Beer app without a check-in mechanism that we can add it in a next release as easy as possible. So, it should be flexible whatever sequence we choose to implement a number of features for the Solid Beer app. We will work with Pano and other semantic modelling experts to finalize the POD data structure design.

And we must also decide if we follow the Untappd way of working for writing and sharing reviews or that we decide on our own simplified approach for the initial release of the Solid Beer app.

Appendix C – Development, Test & Deployment

This appendix will give a highlevel overview of our development environment to develop the Solid Beer app in order to make our results reproducable for others and also maybe to define a first possible best practice for developing Solid app in a certain way.