Clue Analytics

# MVP Requirement – 1.0R0

This document intents to describe the requirement that will cover minimum viable product offering for clue analytics



Author: Walid Darwish

11th of October 2019

Copyright Clue Inc 2019 – Confidential

# Functional Requirements

## Onboarding

### Tenant Onboarding

* Tenant creation must be self-service. As easy as entering the tenant name, contact in formation and credit card info.
* With the tenant creation, a tenant admin user should be created. This tenant admin will have all the abilities to manage all artifacts owned by the tenant. This include, data models management, visualizations management, and user management (including the creation of other tenant admins).

### User Onboarding

* Tenant admins can invite other users to onboard through invitations using only emails.
* After a new user receives an invitation, the process should be self-service.

## Data Connectivity

* Support for database connectivity must include the following database:
  + Oracle and Oracle cloud.
  + MySQL (MariaDB and ClustrixDB?)
  + PostgreSQL (Pivotal GreenPlum?)
  + Microsoft SQL Server and Assure based version
  + IBM DB2 (all variants)

* Support for URL, CSV, JSON, Excel (single sheet) file upload.
* When uploading an Excel file, excel metadata must be respected.
* Uploaded file support must be completely self-service.
* On the first release 1.0R0, only cloud-borne data sources must be supported. A VPN solution must be available for next release.

## Modelling

* UI must give the user the ability to create models based on database connections or uploaded file.
* Heterogeneous data source model support is a must.
* Visual design using DnD for models is a must.
* A single data source artifact in a model can be based on a database artifact (table, view, synonym, stored function, …etc.) in a single connection, an SQL statement, or an uploaded file.
* UI must provide the ability to join different model artifacts. Support for inner and outer joins is a must. The result of a join is always a new artifact.
* UI must provide the ability to filter different artifacts to create new artifacts.
* UI must provide the ability to sort different artifacts to create new artifacts.
* UI must provide the ability to summarize different artifacts to create new artifacts. Minimum set for summarization is Sum, Count, Count Distinct, Average, Min and Max.
* UI must provide Intelli-sense data view on artifacts for the sake of data debugging.
* UI must provide the user the ability to add presentation layer metadata to the model.

## Visualization

* User must be able to visualize data from any artifact in a model or directedly from the data source.
* A single visualization path for both reports and dashboards.
* A visualization contains multiple cards. Each card contains a title and multiple widgets.
* A widget can be:
  + A Chart (minimum support for Bar, Stack, Area, Line, Scatter and Pie)
  + Summary widget (A card with a summarization value, a title and a label)
  + Table; a list representation for a tabular data (model artifact) with formats. Optional summarizations is a must (must be supported but should be optional to the user). minimum set to support is Min, Max, Average, Sum, Count and Count Distinct. Summarization
  + Crosstab with multi-dimensional edges and fact cell values. Summarizations must be multi-level.
  + Formatted text with bullet support

* Client-side sorts and filters must be supported.
* User must be able to publicly share a non-editable version of visualizations or single cards.
* User must be able to preview his visualizations
* Visualizations must be responsive

## Contents

* User must be able to browse contents and share them with other users within the same tenant.
* Complete tenant content isolation is a must.
* Contents must be searchable.

# Non-Functional Requirements

## Presentation

* Zero footprint – runs completely inside a web browser
* Support for Safari, Edge, Firefox, Chrome. Latest long-term support versions all the time.
* Modern and performant UI is a must.

## High Availability

* Product must have 99.999% availability
* Upgrading and expanding the system must not impose interruption on service.

## Globalization

* English language must be supported. Preferably French and Spanish too. If not in the first release, it must be in the one following. When multiple locales are supported, each user should be able to set a contents and presentation locals in his profile.
* A user time zone in the user profile is a must. The user must be able to choose if he wants to see the data as stored or translated to his own time zone per visualization.

## Logs & Monitoring

* Monitoring for the system health status accessible from a super admin account.
* The system should be able to record logs and show the trails of user actions per request.

## Scalability

* The system must be horizontally scalable on all its layers and services.

## Performance

* When running queries against data sources, system should be able to send the most optimized queries and make decisions on what artifacts to process locally based on the context of a single visualization (not a single widget)
* Locally processed visualizations must be distributed whenever possible.

# AI Requirements

* During modelling stage, a trained model will suggest data types whenever missing. For example, when uploading data from a csv file, a data type for each column will be required but not given by the data source. A sampling on the data might be required as an input to the model.
* During visualizations stage, a trained model will suggest the suitable widget type based on the data context of the widget. For example, if I choose a dimension and a measure (product and revenue), the model will suggest bar chart and/or line chart. The suggested widget types will be the ones that will give the most suitable view of the data considering the intended insight of the user.
* During modelling stage, a trained model should be able to give the user modelling hints to create the most performing model. Including the following:
  + Moving filters around to be able to send the filter in a query to the data source to minimize the data processing inside Clue execution engine.
  + Suggesting sorts before joins for joins executing inside Clue engine.