Taskize with Exception monitor

Strategy Deck

TASKIZE

MAKE WORK FLOW

Agenda

Original brief

Defining the problem that we're trying to solve for the industry

Future vision

What prevents us to achieve the vision

Data model and proposed application structure

Solution design

Exception monitor goal

Original brief

The Taskize Exceptions Monitor is a strategic solution designed to streamline the management of exceptions for clients. Currently, users rely on manual processes to track and resolve exceptions across various systems, leading to inefficiencies and inaccuracies. **This initiative aims to integrate exception data directly into Taskize, allowing users to import exceptions via flexible methods (such as flat files, API, or direct integration)** without needing transformation.

Once exceptions are uploaded, users can visualize, update, and automate them within Taskize, linking exceptions to Bubbles. **This connection ensures real-time updates between exceptions and Bubbles, reduces manual workload, and generates accurate analytics, such as time-to-resolution and operational cost metrics.**

Taskize Solution Proposal - Exceptions Monitor

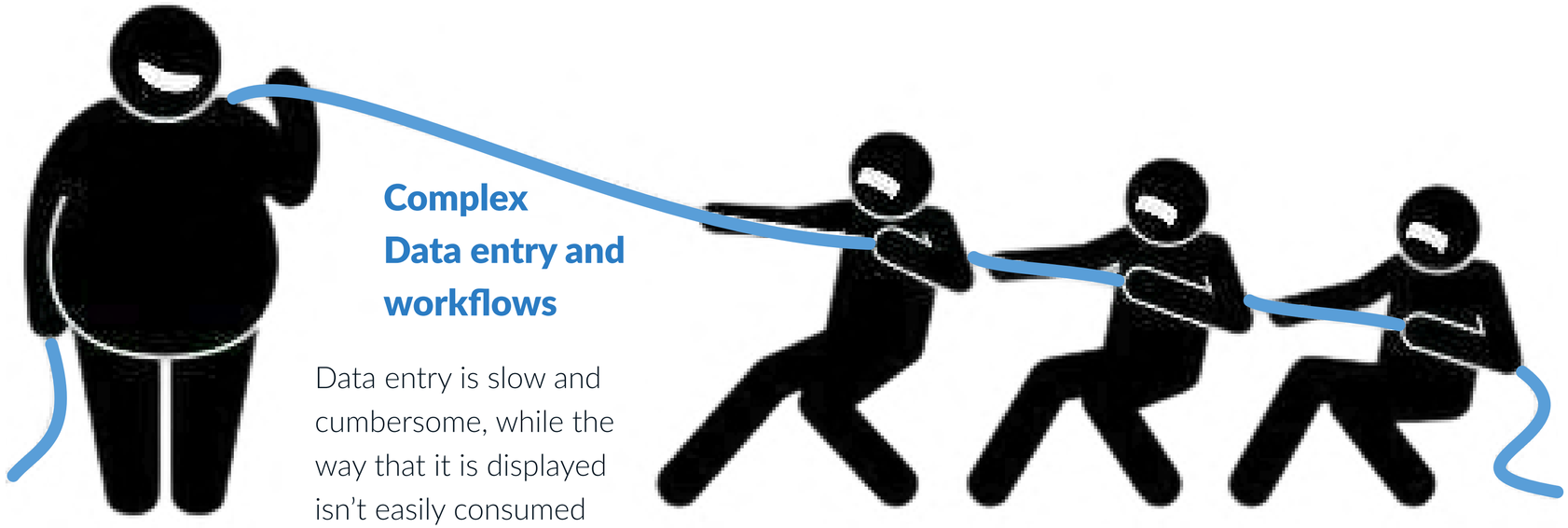
Download brief here

Conflicting goals

Defining the problem that we're trying to solve for the industry

What we promise our customers

Structured data, that can easily be organizable, searchable, reported on and with faster resolution times



Complex Data entry and workflows

Data entry is slow and cumbersome, while the way that it is displayed isn't easily consumed by users

What are the goals of the actual users

Users are evaluated as per amount of queries that are successfully solved as quickly as possible.

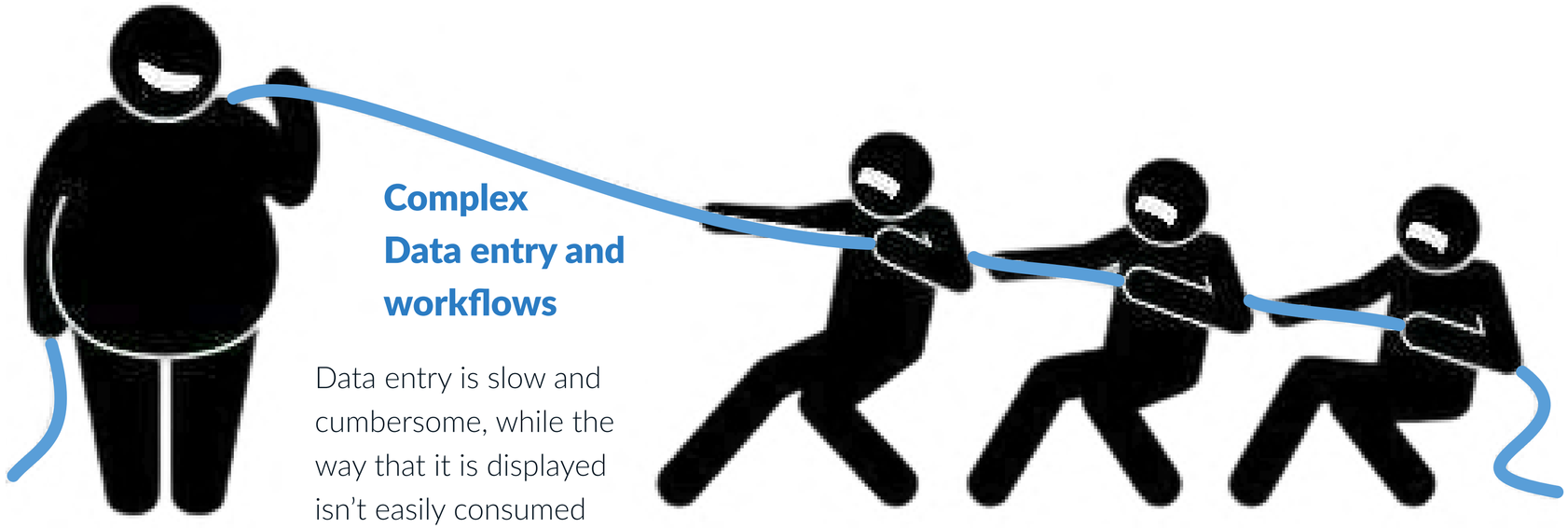
Conflicting goals

Defining the problem that we're trying to solve for the industry

For **Taskize**'s exception monitor to be effective, it **must present data from multiple sources in a consistent and easy to consume format**. This ensures that users can quickly identify priorities while benefiting from enhanced usability features and other automation capabilities that streamline and improve the accuracy of data entry.

What we promise our customers

Structured data, that can easily be organizable, searchable, reported on and with faster resolution times



Complex Data entry and workflows

Data entry is slow and cumbersome, while the way that it is displayed isn't easily consumed by users

What are the goals of the actual users

Users are evaluated as per amount of queries that are successfully solved as quickly as possible.

Typical process from exception raised to settlement

Defining the problem that we're trying to solve for the industry

User receives Financial issues* from a multitude of tools

User selects single issue that needs to be solved

User groups multiple issues into a single wrapper view in another tool

XLS

User contacts relevant counterparty to help solve exceptions, via preferred method of communication

Users from both counterparties use several tools to confirm information while communicating amongst themselves to solve the problem

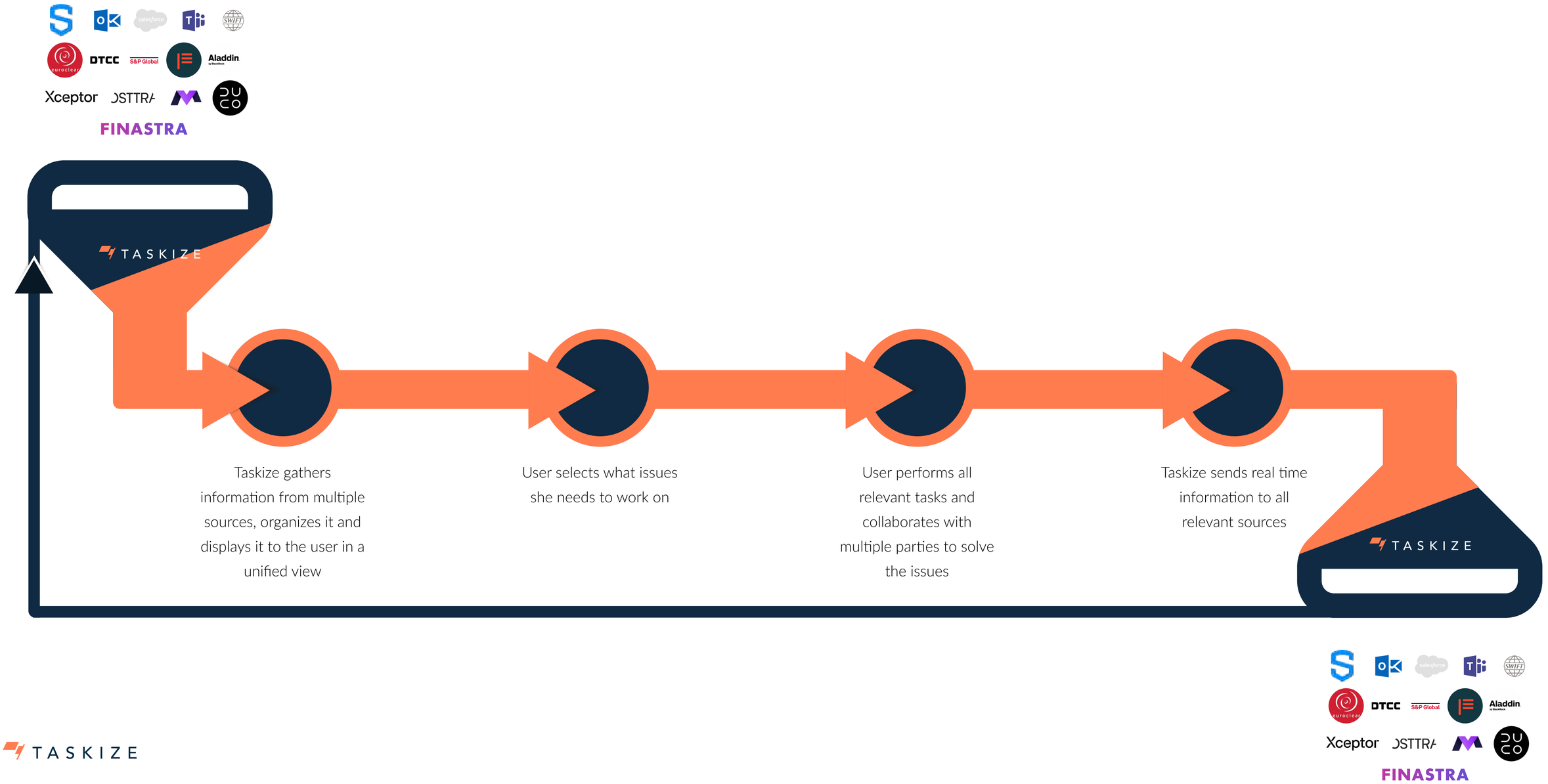
Issue is solved

TASKIZE

* An issue can be an exception, an onboarding request, portfolio reconciliation, margin disputes, predictive settlement issues. etc.. It can be whatever the user needs to solve.

The vision for an improved process using Taskize to all it's capabilities

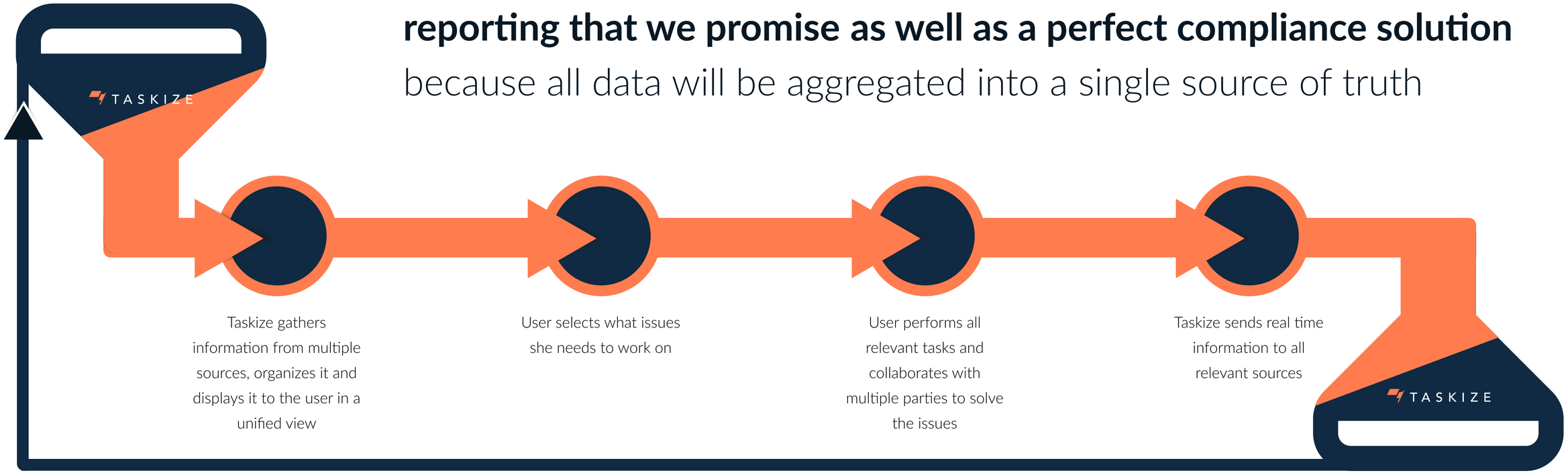
Future vision



The vision for an improved process using Taskize to all its capabilities

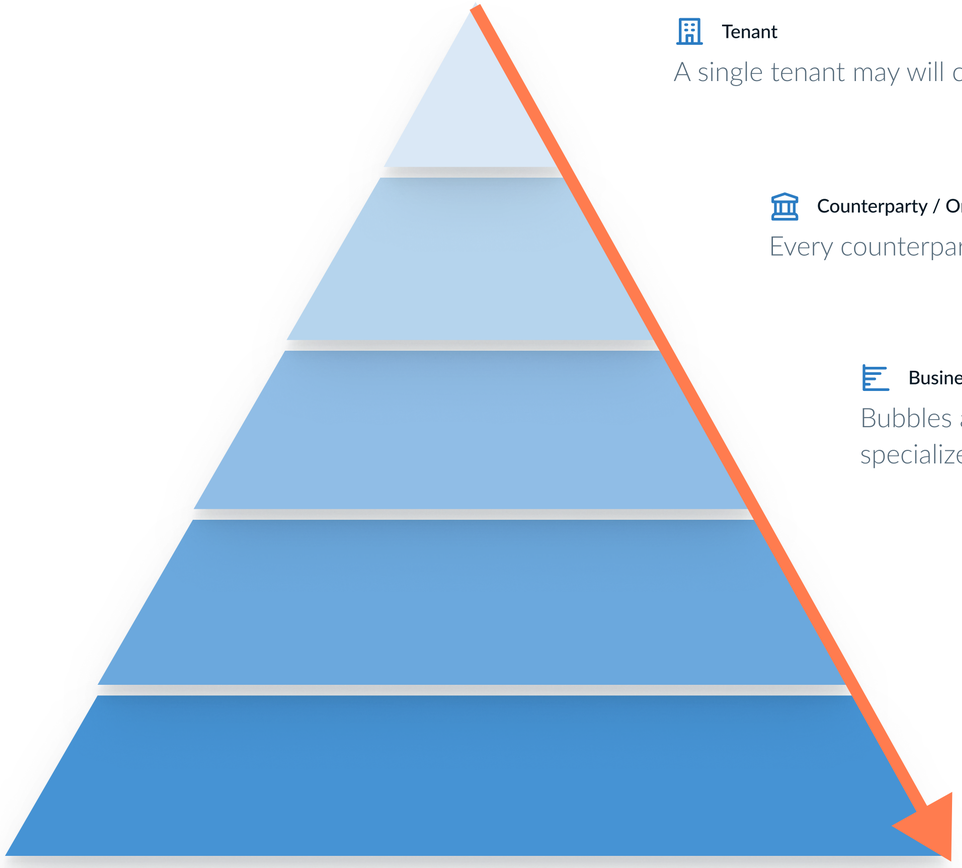
Future vision

By making Taskize as an aggregator of financial requests, we can make users much more efficient and **allow our customers to achieve their T+1 goals**, while we're also able to **provide them with the analysis and reporting that we promise as well as a perfect compliance solution** because all data will be aggregated into a single source of truth



Current data model

What prevents us to achieve the vision



Tenant

A single tenant may will consist of one or more counterparties or organizations.

Counterparty / Organizations

Every counterparty will consist of one or more business units and functions.

Business Units / functions

Bubbles are organized by business units, which can encompass a whole team, a team segment, a specialized skill set, a location, or even the users responsible for customer interactions.

Bubble

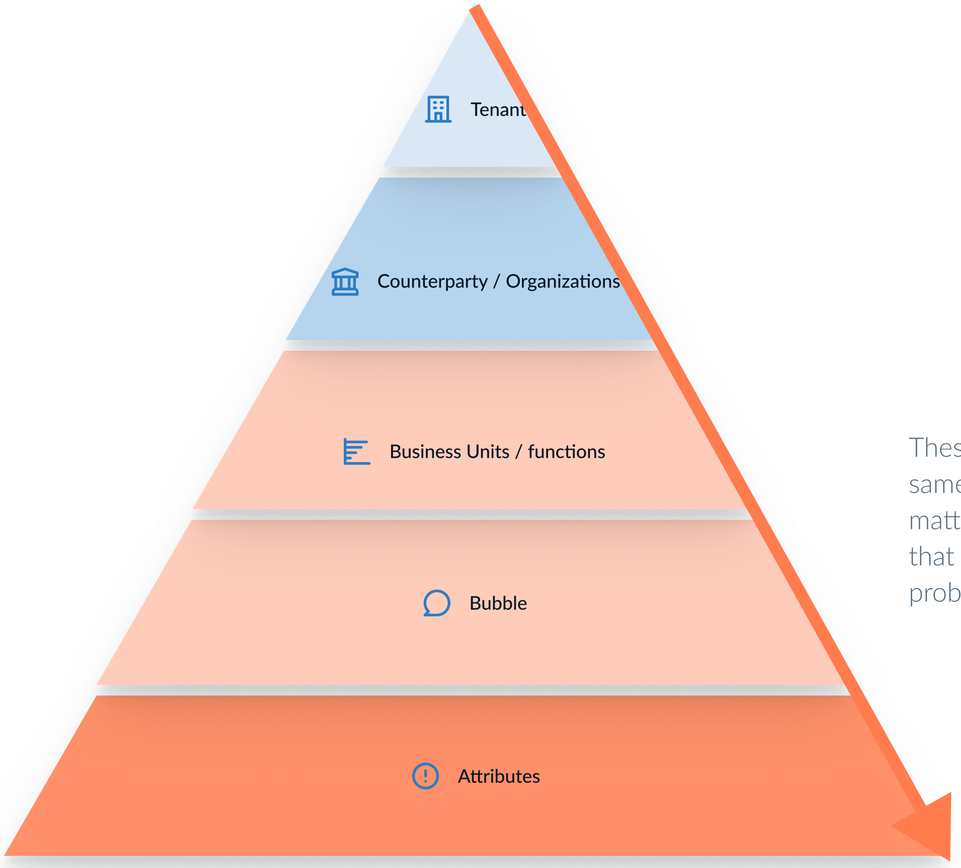
Bubbles serves as our central hub for resolving issues; however, they can be simple internal conversations, without referencing any exception, they can be portfolio views with a single conversation, they can be an exception, they can be internal conversations about other exceptions that are located on another Bubbles. Effectively users are nothing more nothing less than conversations about anything and everything… it's hard for users to focus on what is really important and easily parts of the problem are scattered in multiple bubbles without an obvious connection.

Attributes

Attributes serve as key data points that allow for the swift identification of issues, leading to quicker resolutions and enhanced reporting capabilities. However, users don't usually fill in attribute information because of the time it takes to add said information and the lack of features (NLPs, OCRs, Bulk imports, direct API connectors to other tools…) that would automate the filling of attribute information.

Current data model

What prevents us to achieve the vision

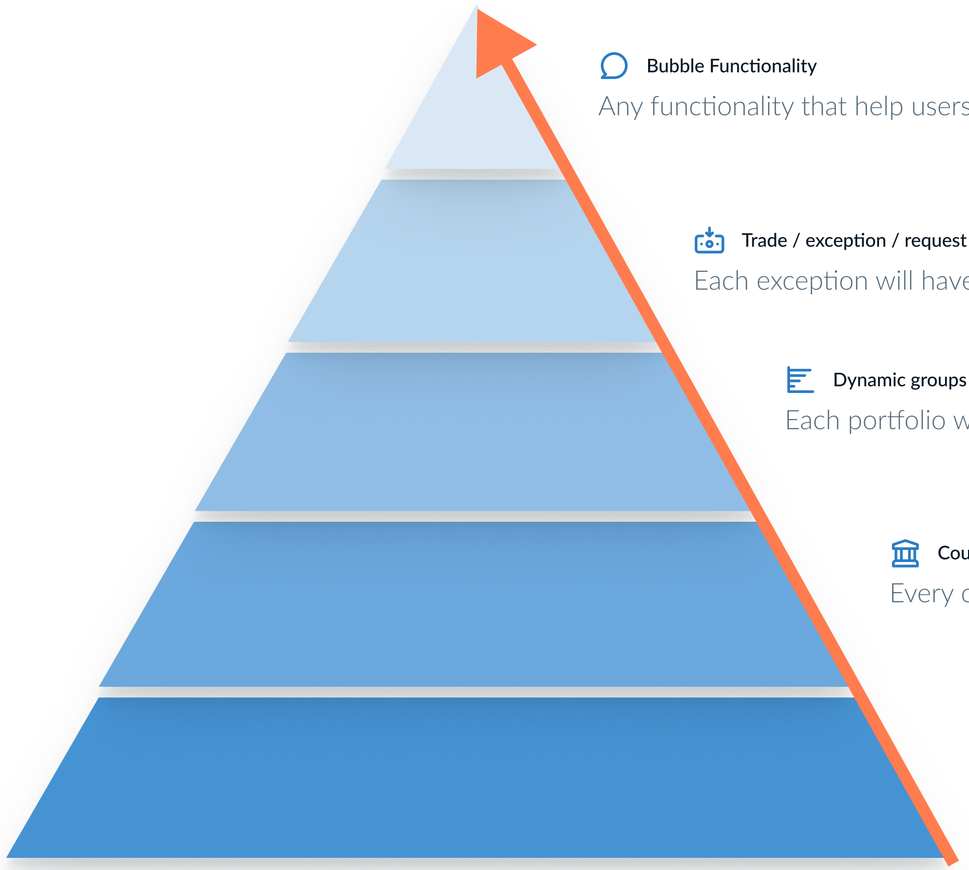


These sections can be so many things at the same time that they're confusing. To make matters worse, there isn't an actual structure that would support users to identify and solve problems in a quicker and more effective way

Attributes aren't being filled which also make it harder for users to solve their issues

Proposed data model

Solution Design



Bubble Functionality

Any functionality that help users to solve the issues, journal, attachment reviews...

Trade / exception / request view

Each exception will have 1 or more Bubbles associated to them

Dynamic groups

Each portfolio will have 1 or more exceptions

Counterparty / Organizations

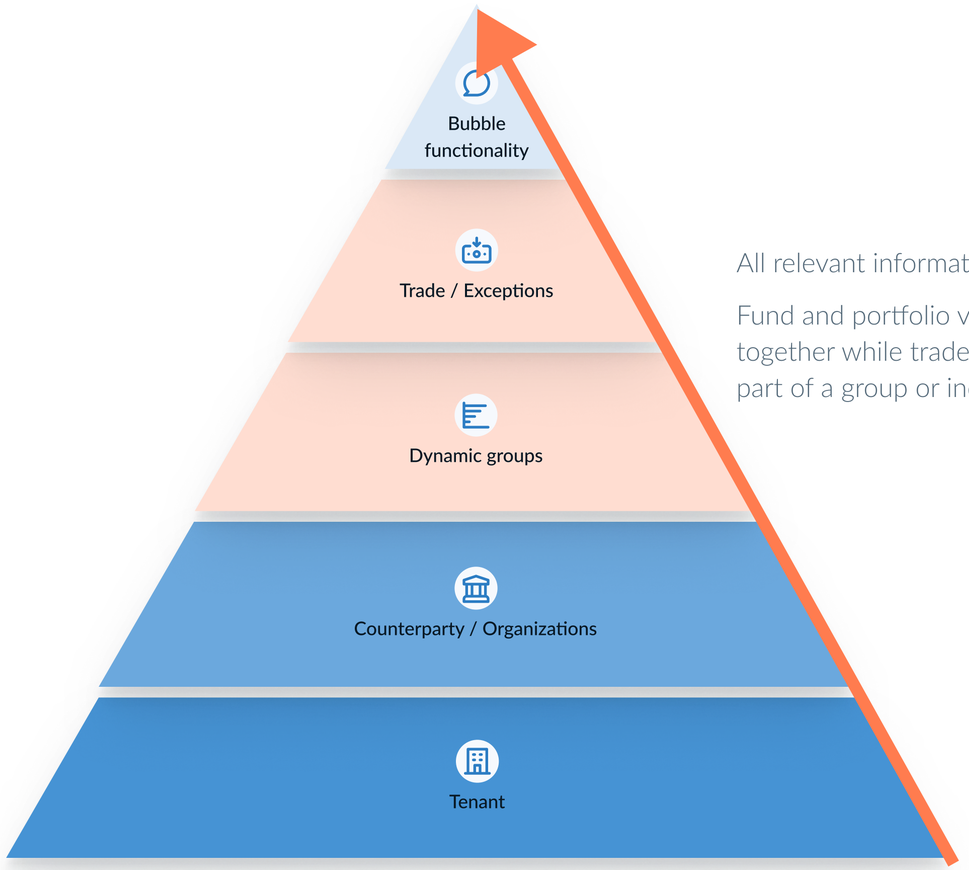
Every counterparty will include of one Portfolio or Trades / Exceptions

Tenant

A single tenant may will consist of one or more counterparties or organizations. / Barely any information is available about this level of information within the platform

Proposed data model

Solution Design



All relevant information to solve the issue should be seen here.

Fund and portfolio views are multiple issues that can be grouped and solved together while trades or exceptions are single issues, which can be solved as part of a group or individually

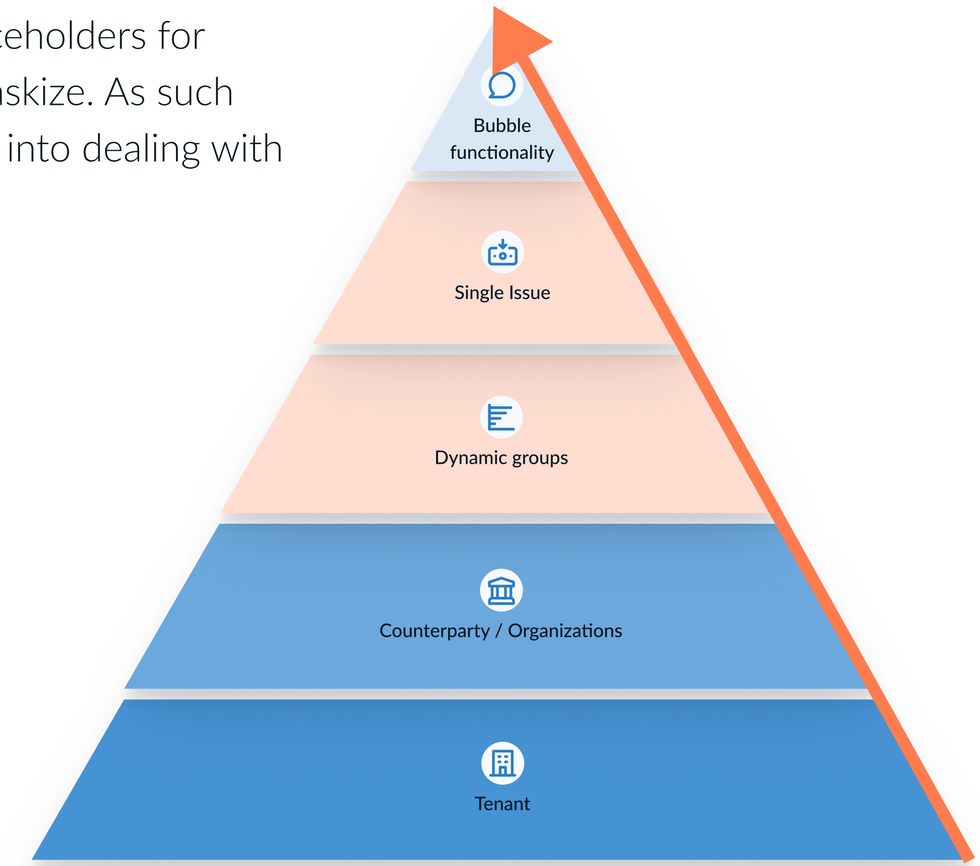
Differences between the data models

Solution Design

Bubbles and Business units are basically placeholders for anything and everything that exists within Taskize. As such there's no mental model that supports users into dealing with their daily tasks.

No logical structure

≠

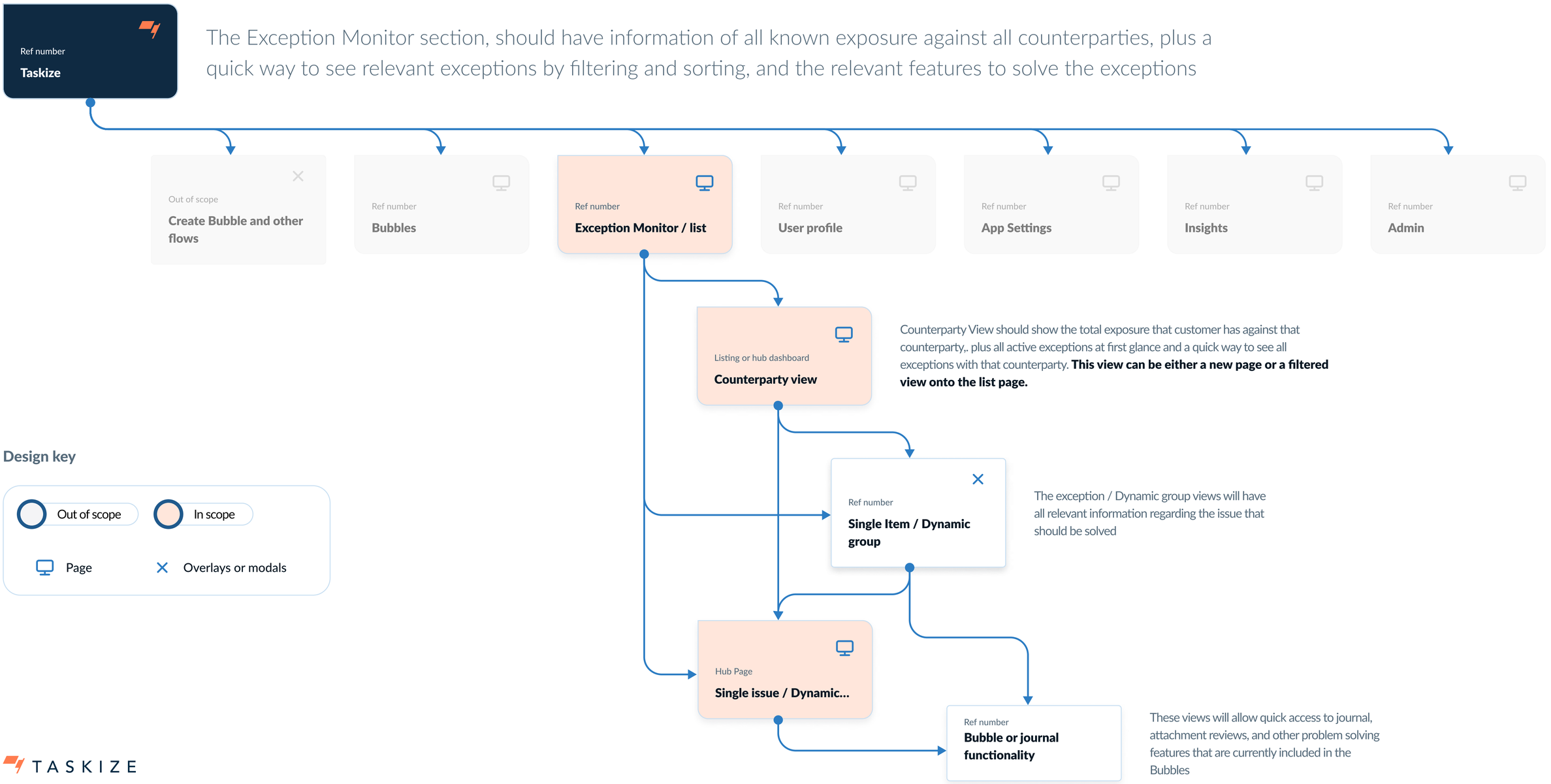


Structure that leverages current user behaviour to create meaningful relationships between elements

Information architecture for exception monitor

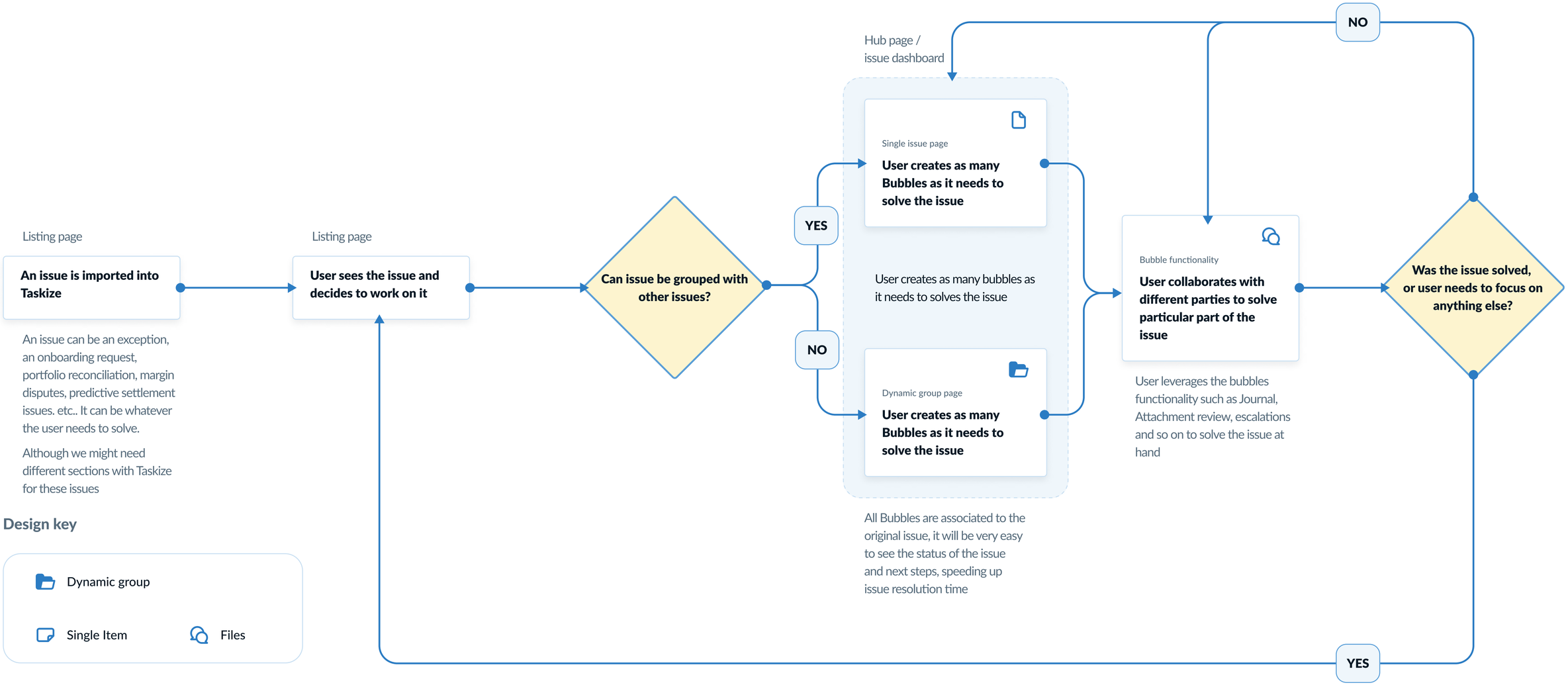
Solution design

The Exception Monitor section, should have information of all known exposure against all counterparties, plus a quick way to see relevant exceptions by filtering and sorting, and the relevant features to solve the exceptions



Process of solving issues in new data model

Solution design



Possible features of and capabilities for the success of Taskize

Solution design

Smart Form Design & UX Optimization

Minimalist UI: Reduce the number of required fields and steps.

Inline Validation: Show real-time validation instead of waiting for form submission.

Keyboard Shortcuts & Tab Navigation: Allow power users to navigate forms without using a mouse.

Auto-Fill & Auto-Complete: Predict user input based on past data or suggestions.

Bulk Data Entry: Let users input multiple records at once.

Multi-Step Forms with Auto-Save: Prevent data loss in case of accidental navigation.

Automation & AI-Powered Features

Predictive Text & Smart Defaults: Suggest values based on past entries or patterns.

AI-Based Error Detection: Highlight inconsistencies or incorrect values in real time.

Auto-Categorization & Tagging: Use AI to classify and tag data automatically.

OCR (Optical Character Recognition): Convert scanned documents into structured data.

Pre-Filled Templates & Reusable Data Snippets: Allow users to save and reuse common entries.

Collaboration & Role-Based Access

Real-Time Collaboration: Multiple users can enter or edit data simultaneously.

Role-Based Permissions: Customize who can enter, edit, or approve data.

Integration & Connectivity

APIs for Data Sync: Enable seamless integration with CRMs, ERPs, and other third-party tools.

Database & Spreadsheet Imports: Allow CSV/XLS uploads and smart parsing.

Performance Optimization

Fast Load Times: Optimize backend processing for real-time responsiveness.

Progress Indicators & Feedback: Show users loading states and success messages to keep them informed.

Onboarding case solution - Build Requirements feature list

Solution design

Feature	Detail	Phase 1	Phase 2	Phase 3	Phase 4
Ingestion from 3rd party sources (Exception manager recycle)	Partner platforms to integrate into Taskize (eg. Saphyre, Doji, S&P OA)	X			
	Automatically associate work items with the same onboarding request	X			
Taskize Capability	Support exception management view for onboarding requests	X			
	Determine and map jurisdiction risk against countries	X			
Ingestion from email	Automatic routing/assignment of an onboarding email		X		
	NLP capabilities to auto-assign attributes		X		
	Normalise data inputs to support multiple sources in a single view			X	
Data enrichments	SSI feeds (T+1 recycle/Ssimple opportunity)				X
	NAV (Cpty Mgr etc recycle of S&P API)				X
Encompass lookup	Support offboarding checks				X

Nuno Viegas

nuno.viegas@taskize.com

UX and product design lead

TASKIZE

MAKE WORK FLOW