



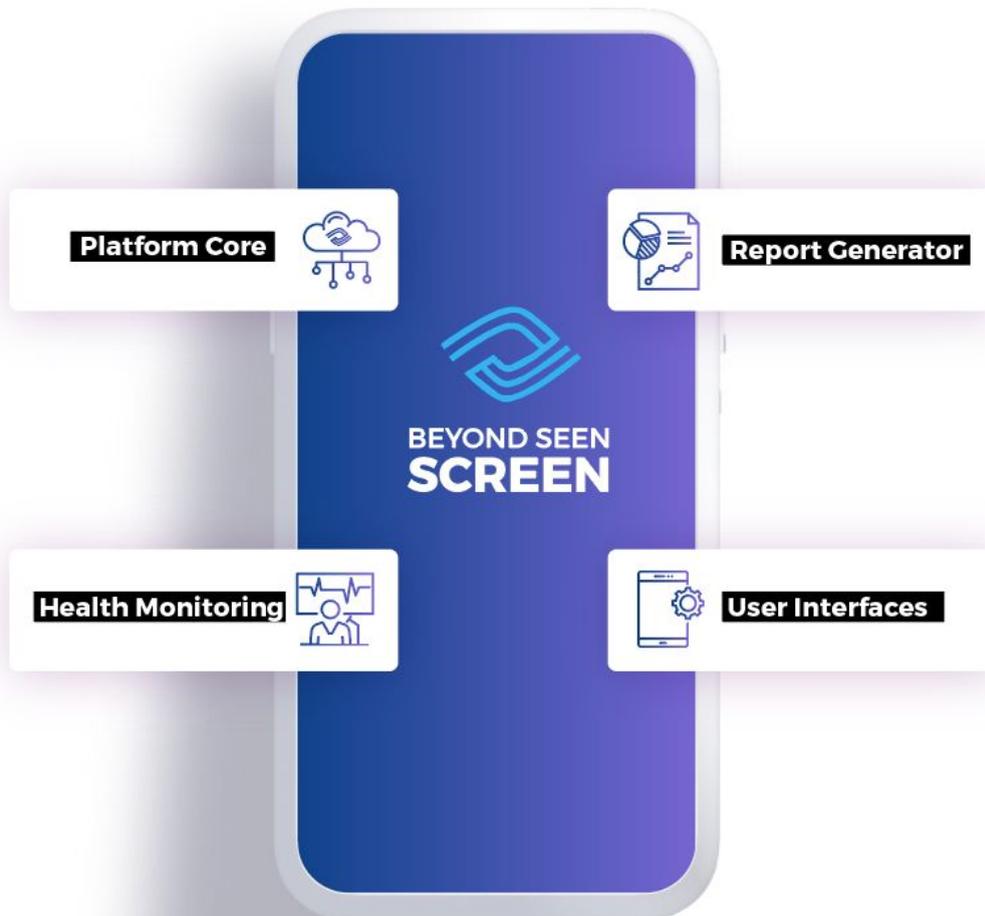
TECHNICAL WHITEPAPER

Platform overview

Beyond Seen Screen is a platform that allows users to scan the video content they are watching with their smartphone and receive additional information related to that video. The information that is returned to the user is configurable by the content owner.

The platform is not just an idea. It is currently a working product! A working version of the platform is already implemented and hosted in the cloud. Existing implementation of the Beyond Seen Screen platform consists of the following modules:

- Platform core
- User interfaces
- Health monitoring
- Report generator



Platform core is the central part of the platform implementing all of the core business logic. Its modules serve data to the user interfaces. A separate standalone system monitors the health of the platform. Generating business reports is done in a standalone report generator system.

Platform core

Platform core is the part of the system that does the following:

- Stores the platform data
- Executes the platform logic
- Prepares the data for presentation to the users

Platform data storage includes a mix of technologies with the aim to optimize resource use and performance of the system. SQL database and separate object storages are used for different types of data. Special care is given to security aspect of data storage with the main aim being reducing exposure of data to the rest of the system as well as to the public.

Platform logic executing part of the platform has a modular architecture, allowing easy scaling of different parts of the system to meet the load requirements. Modular architecture also allows rolling out new features (like different ways to process platform data) to a selection of users in a controlled way.

Separate platform modules are used to prepare the data for presentation to different types of platform users through dedicated APIs. Such approach allows independent scaling and availability management of different platform parts in accordance with the changing load on the system from different types of users.

Platform core is implemented in .Net Framework using C# programming language. A group of virtual machines are running different modules of the platform core.

User interfaces

The two main types of users the platform will serve the data to are the clients and the viewers. Clients will prepare the different experiences for the viewers, who will consume them on their mobile devices.

Client perspective

The platform's web application provides clients with an easy to use and modern UI to accomplish all tasks required to prepare the additional content they want to present to viewers. The existing UI leverages a combination of server side scripting and JavaScript to provide all of the tools needed to get the job done.

Viewer perspective

Viewers can use their Android powered (Android 5.0 and later) mobile devices today to get additional information about the video content they're watching via native app. Support for iOS (version 9 and later) is planned and included in the project roadmap.

Health monitoring

Platform status is monitored by a standalone system. The level of information detail provided by the system depends on the type of user requesting the data. The general public can see if all of the platform main modules are working or not while the platform administrators get a more detailed look into the whole platform with details like the level of load on a particular platform mini-module. Until the platform is improved to handle scaling resources to meet the system load in a fully automated way, the health monitoring system will be used to allow platform administrators to understand if the platform's performance would benefit from extra resources.

Health monitoring is implemented in .Net Framework using C# programming language.

Report generator

At the end of the day, the platform needs to provide added value to their users to keep users coming back to it. While the viewers will easily know whether they're getting value from the platform, the clients need to be provided with different reports. An extensible report generator module is implemented in a form of web services to create these reports without affecting the performance of the rest of the platform. The report generator is configured from a web interface and runs on a schedule producing all of the reports the clients will be looking for.

Report generator is implemented in .Net Framework using C# programming language.

Features

Current feature set

The platform currently implements the following features that allow complete real life usage of the project:



Core functionality

Scan the video using an Android app, have the platform recognize it and return the additional information which gets displayed in the Android app.



Administrative portal

A web app that allows administrators of the platform to configure the content the platform recognizes and the data it returns.



Health monitoring

A web app that provides a status page showing the health status of the platform.



Report generator

Web service that allows configuration and automated creation of business reports from platform data.

Future feature set

The platform will be improved in multiple directions, offering improvements to viewer experience, configuration ease, value it provides to clients and improved automation of platform maintenance. The schedule on which the features will be released might be affected by the amount of funds raised, but the list of features in the development queue can only grow but will consist of at least the following:



Customer portal

A web app allowing clients to manage their content and additional information on their own.



Social features (sharing)

Ability for viewers to share the experiences they receive from the platform with the rest of the world.



Shopping

Improved shopping experience on the platform through faster access to items and services displayed using a unified look and feel in the native mobile app.



iOS support

A native iOS app to bring the platform benefits to our Apple using viewers.



Analytics

Ability for clients to get live configurable insight into platform use data.



Geolocation

Viewers will be able to share their geolocation with the platform to allow the platform to display more relevant information.



Personalized information

Viewers will get the ability to share their data like gender, age and preferences with the platform and in return receive more personalized information.



Custom interface layouts

Support for custom interface layouts will be implemented, allowing clients to choose a different and unique way to display the additional information on their videos.



Screen detection

Smartphone apps will be extended with technology that detects the screen that displays the video content, making it even easier for viewers to scan the content they're watching.



Live channels

The platform will be improved to work not only with the prescanned video content, but also live TV channels and allow the smartphone to become a true second screen.



Smartphone screen as input

Instead of using the smartphone camera to look at the video the viewer is watching on another screen, we will allow the application to look at the smartphone screen and see what the viewer is watching on that screen.

Platform will now be ready to provide service to viewers who use their mobile devices as their primary screen on which they enjoy their streamable and video on demand services like Netflix and Hulu.



TV integration

Apps for smart TVs will be created, bringing Beyond Seen Screen natively into TVs. Wouldn't it be nice to have a BSS button on your TV remote?

API

The modular architecture of the platform implementation requires use of the APIs between all platform modules and inside the data pipelines used within the platform. Having these APIs implemented from the project start allows us to form partnerships with companies looking to integrate our technology into their product or those that offer technology that could improve or augment our product. Modules like video recognition, data maintenance (storage, insertion, editing), data querying, health monitoring, report generator, client and administrative portals all leverage the internal APIs to talk to the rest of the platform. Customized user interfaces can be built for clients or by clients who can then leverage their internal software tools and processes to streamline maintenance of the data in Beyond Seen Screen platform.

Blockchain

Beyond Seen Screen will use the blockchain to power its business transparency. Instead of recording platform usage in the private database, Beyond Seen Screen will save that data into a smart contract on the Ethereum network. By doing that, Beyond Seen Screen will build trust with clients that the data Beyond Seen Screen uses to create invoices for its service is true and unaltered.

The smart contract that Beyond Seen Screen will utilize to record the platform use will be implemented in Solidity programming language and will be deployed to and executed within the Ethereum Virtual Machine. Technical prerequisites to implement this smart contract already exist in the Ethereum Virtual Machine and the Solidity programming language. In its essence, the smart contract will keep a mapping between the ID of the video content (same ID assigned to the video in Beyond Seen Screen's private database) and the number of scans that video received. Whenever the platform recognizes that a viewer scanned a particular video from its database, a smart contract function will be called which will increase the value associated with that video. The time when the scan occurred will be inferred from the date and time when the Ethereum block that records this smart contract function call was mined.

Care will be taken to keep the cost of running such smart contract under control. Historically, the average price of an Ethereum transaction was anywhere between 1 US cent and 4 US dollars. If Beyond Seen Screen plans to have millions of scans per day and if it plans to record each scan on the blockchain in it's own transaction as it happens in real time, it will surely not be able to pay

those fees for too long. Cost optimizations will have to be implemented. One obvious optimization is to allow the smart contract function to accept a number by which it should increase the scan count and then for the platform to collect scanning events for a configurable time period (a few seconds or minutes) and then increase the scan count in the smart contract for every video by the amount of actual scans. If a single video was scanned for 5 times in a configurable time period, then the platform makes only a single call to the smart contract (single transaction) to record 5 video scans. Another optimization is to extend the smart contract function to accept a list of video IDs and the number by which to increase their scan count in a single function call. That way, if in a given configurable time period there were 10 videos scanned, we can record information about all 10 videos in only a single transaction, cutting the costs down further. The last obvious optimization is to extend the period in which the platform collects its data use before it writes it to the blockchain, up to a full day. That way, the information about all videos scanned in a single day with their respective scan counts can be written to the blockchain in a single transaction. And the clients can rest assured that after the data for a full day was written to the blockchain, it can not be altered any more.

Hosted on Amazon Web Services

The platform is hosted in the Amazon Web Services (AWS) Cloud. The AWS Cloud provides a broad set of infrastructure services, such as computing power, storage options, networking and databases that are delivered as a utility. New services can be provisioned quickly, on demand and available in seconds. Today, AWS provides a highly reliable and scalable infrastructure platform in the cloud that powers hundreds of thousands of businesses in 190 countries around the world. This allows Beyond Seen Screen to access the building blocks needed to respond quickly to changing business requirements.