The Sahara Framework: For Blockchains

Kazimir Dugandzic
blockchain.saharaframework.com
www.innvtv.com

Abstract
A generalized framework for building and deploying multi-tenant SaaS applications on the Azure Cloud with transactional and smart contract layers managed on decentralized public/private/ hybrid blockchain networks.

1. Introduction

Sahara Framework allows you to quickly compose azure resources and blockchain infrastructure to build out highly scalable applications with distributed ledgers. Development time is accelerated using templated solutions and components along with build/release pipelines for Visual Studio Online.

2. Use Cases

Use cases include hybrid centralized/decentralized applications with settlement layers on distributed ledger(s) such as Bitcoin, Litecoin and Ethereum. These applications can integrate with layer 2 networks and side chains such as Lightning and RSK along with private consortium chains.

Participants use desktop and mobile apps to work with decentralized data and assets using a SaaS based portal on the Azure Cloud supported by centralized services such as Azure Service Fabric, Azure SQL Server, Cosmos DB, Azure Search, Azure ML, etc…

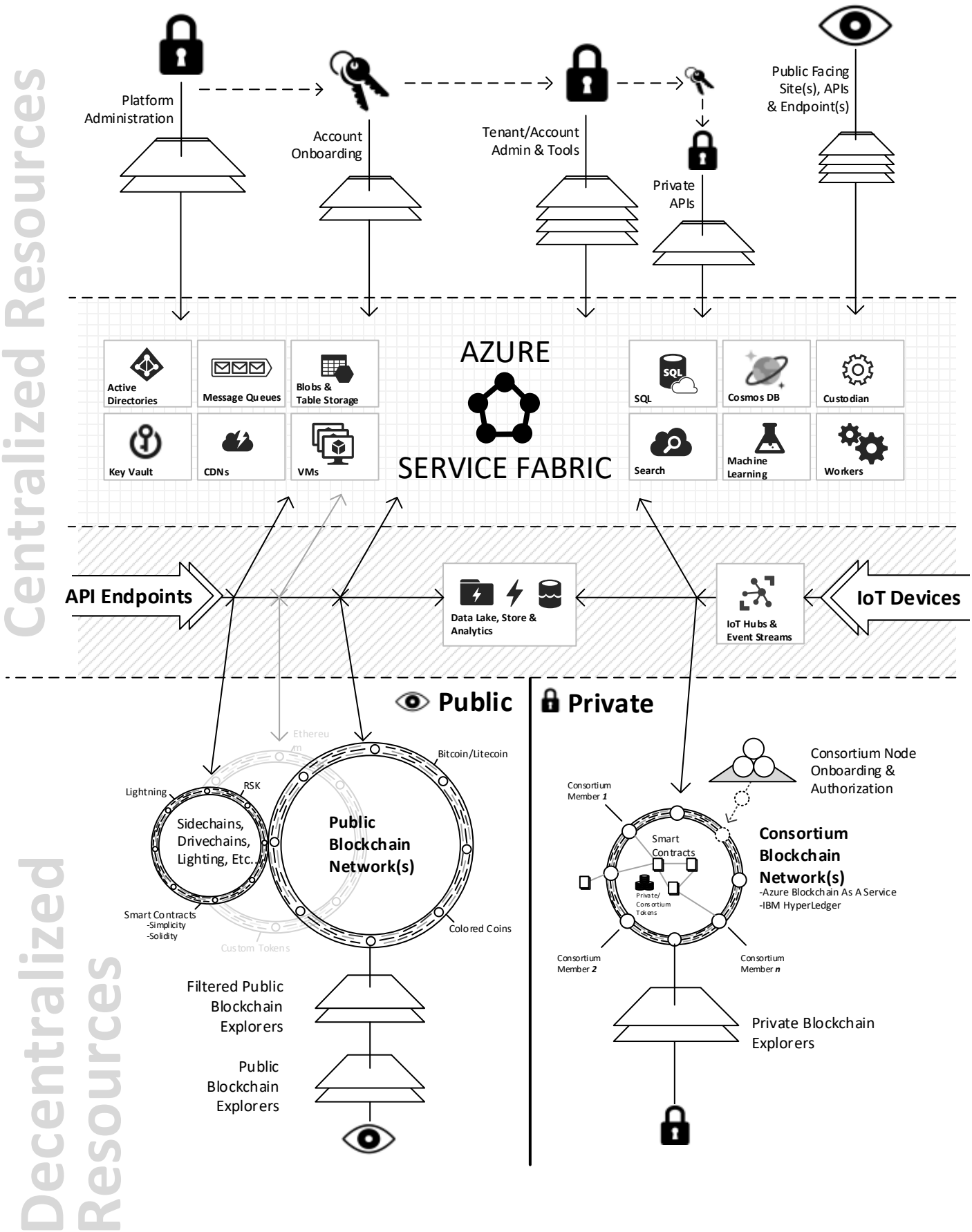
3. Scalability

Built to take advantage of Azure scalability by leveraging Service Fabric, VM Scale Sets, IoT Hubs, Event Streams, Data Lake, CDN Endpoints, Cosmos DB, Table Storage Message Queus and more.

4. Architecture *(See figure A)*

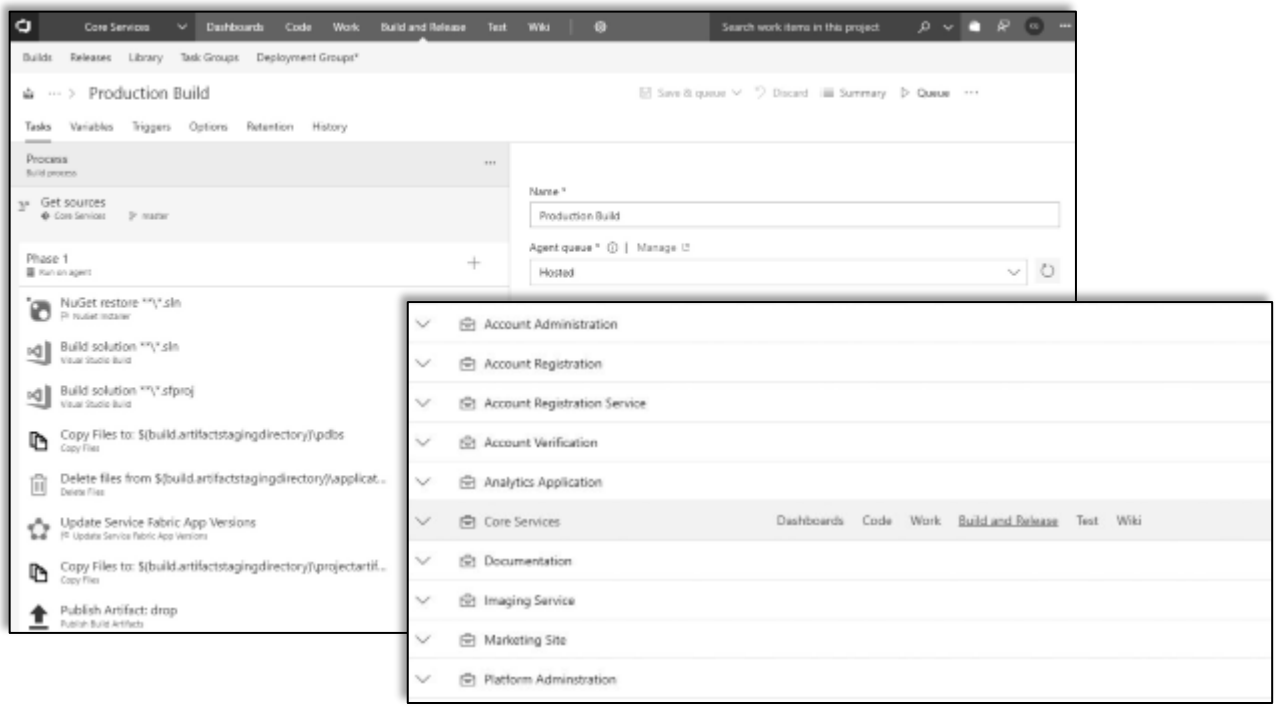
A micro-services based architecture allows you to separate concerns across multiple teams with multiple skillsets. This give you the ability to develop components in parallel paths and makes future upgrades, versioning and product pivots easy to manage.

Figure A: *The Sahara Framework high level architecture*



5. Development & Lifecycle Management
Visual Studio Online Team Services is used to manage code repositories as well as fine tune the test, build and release process.

Figure B: *Code repository, test, build & release management for Sahara Framework within VS Team Services*



6. Micro-Services Architecture
Leveraging Azure Service Fabric, it is easy to upgrade and extend your platform with micro service components with less risk and quicker time to market.

7. Production Scenarios
Sahara can be deployed in a variety of ways. It can be publicly available for multiple tenants to generate accounts (a typical B2C/SaaS scenario), a private B2B extranet requiring an invitation for onboarding or as an internal platform with minimal public endpoints.

8. Centralized Accounts
Accounts that require centralized control are stored in SQL and can also take advantage of Azure Active Directory as well as Azure AD B2C.

9. Centralized Security
Custodial private key storage takes advantage of the Azure Key Vault Service.

10. Decentralized Accounts
Accounts can be generated outside of the centralized infrastructure in a private manner within the blockchains (public, private or consortium) using self generated public/private key pairs. When needed these can be stored in the centralized system as a custodial service, otherwise control of the private keys can remain in the hands of the account holder and can remain offline.

11. Smart Contract Management

Create & manage smart contracts using individual, account and platform views. Smart contracts can live on the Lightning, RSK, Ethereum or any number of other platforms. Built in tools allow for uploading and deploying Solidity, Simplicity, Ivy, Bitcoin Script or any other blockchain scripting language.

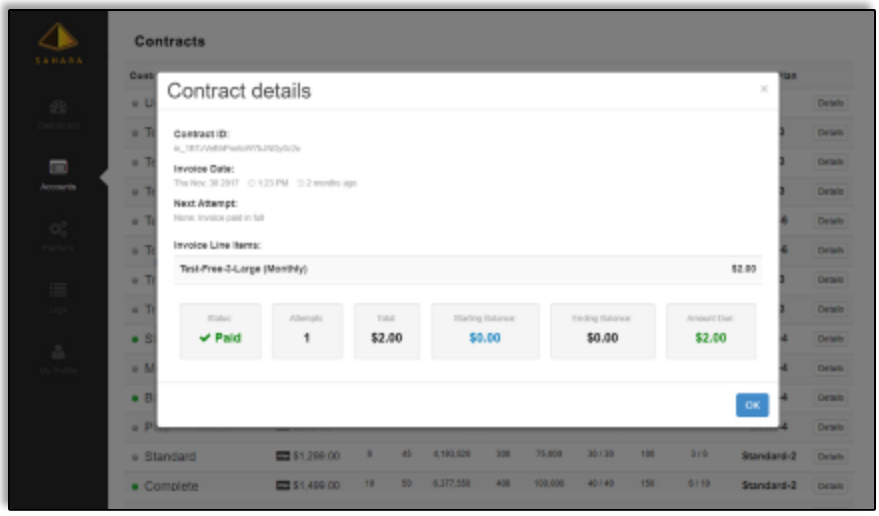


Figure D: *Contract manager within the Platform Administration Portal.*

12. Wallet Management

Create & manage wallets and funds on centralized portals, or offer offline management capabilities via service layers or direct access to the public or underlying private/consortium blockchains/sidechains.

13. Block Explorers

Custom block explorers can be integrated which allows filtering of blockchain noise.

14. Transaction History & Event Logs

View transaction history platform wide on the administration portal(s). Historical data can live on the blockchain as well as in centralized storage for fast access with verification against the underlying blockhain data. Detailed and searchable event logs can also be kept off-chain for a more granular look at transactional history.

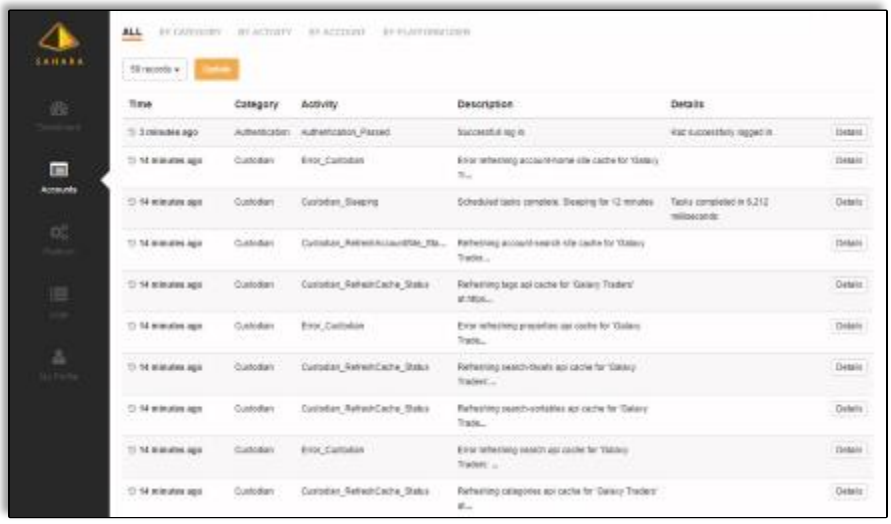
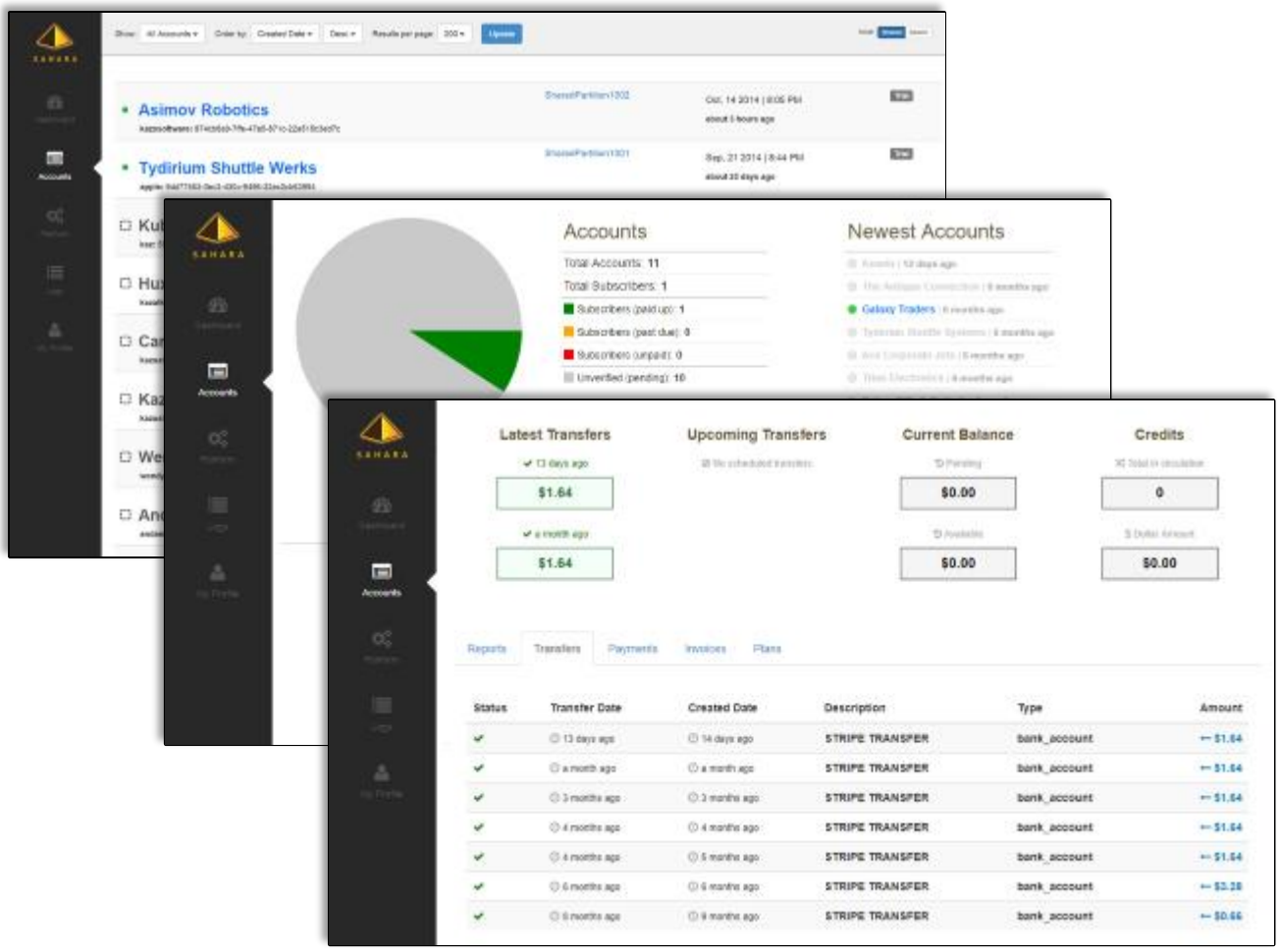


Figure E: *Event log viewer within the Platform Administration Portal.*

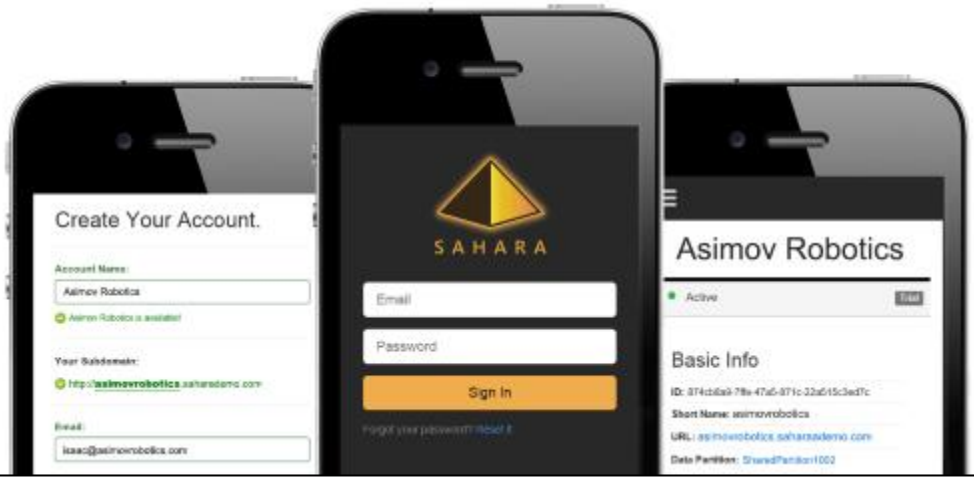
15. User Experience

UI templates make creating administration portals that plug into the underlying infrastructure fast and easy.



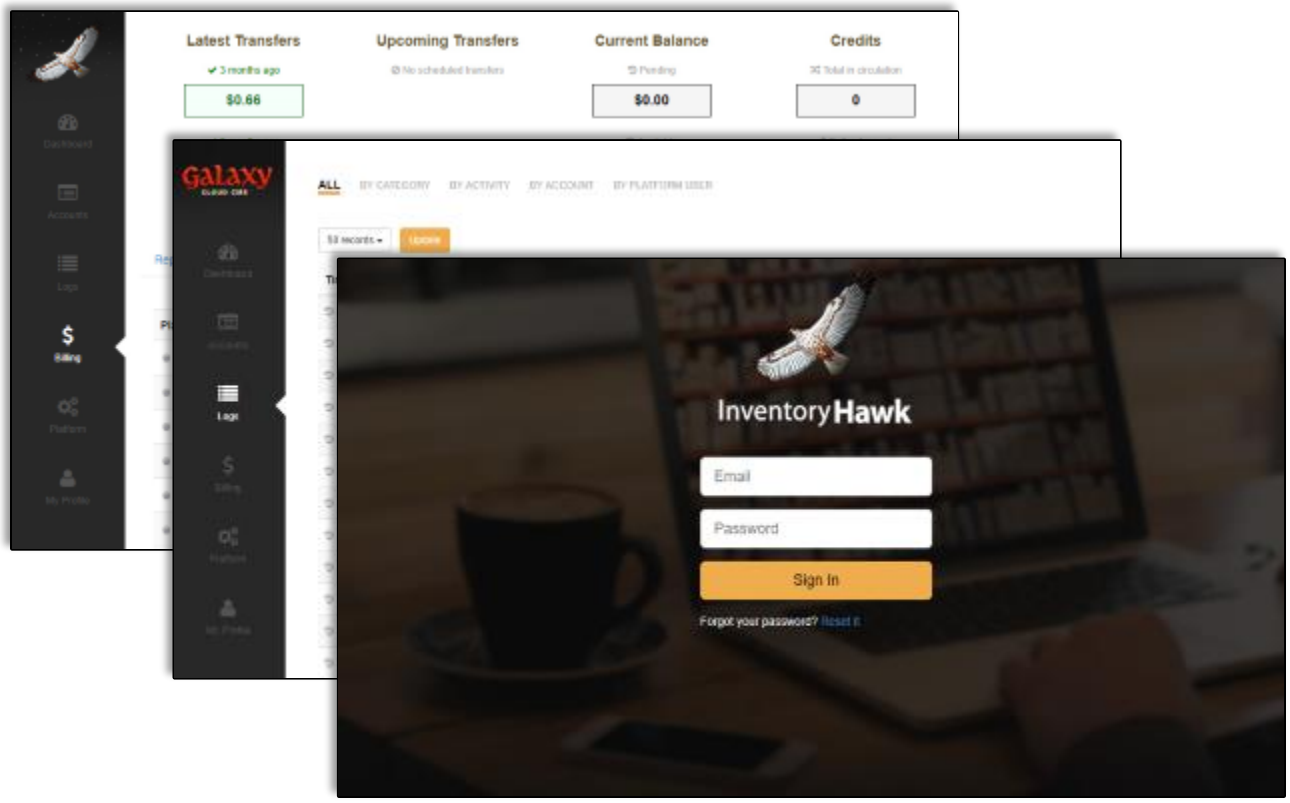
16. Responsive/Mobile Views

Bootstrap allows for responsive views out of the box



17. Branding

Branding the UI to your business is also a simple matter. This frees you up to focus on making your application powerful, secure and unique.



---

18. Conclusion

This is just a small snapshot of what is capable with the Sahara Framework. Our aim is to be as general as possible in order to allow ANY application to be built in a scalable way on both centralized and decentralized infrastructure with a robust IoT and API services layer that allows for an unlimited amount of integration points.