# 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.



# 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 1	3: Code repository,	test, build	🗢 release mana	gement for Sahari	a Framenork	within V	/S Team.	Services
- <del>-</del> -	· · · · · · · · · · · · · · · · · · ·			S				

🗘 Core Services 🗸 Dashbourds Code Work Buil	id and Release Test Wiki   69 Search vor	ark itema in this project 🛛 🖉 🔍 🚥
luilds Releases Library Task Groups Deployment Groups*		
a ··· > Production Build	🖂 Save & queue 🖂 🦻 f	Discard ill Summary D Queue
uks Variables Triggers Options Retartion History		
hocasa Id prezo		
Get sources © con Service  P mater	Name * Production Build	
Phase 1 Il non agent	+ Agent queue * ()   Manage (3 Hosted	~ 0
NuGet restore **\*.sin P nudet indaw	Account Administration	
Build solution **\*.sln Your State Built	✓	
Build solution **/*stproj	V 🐵 Account Registration Service	
Copy Files to: S(build.artifactstagingdirectory)/updbs Copy Files	✓ ( <sup>1</sup> ) Account Verification	
Delete files from \$(build.artifactstagingdirectory);applicat	✓	
Update Service Fabric App Versions P Update Service App Versions	✓	Dashboards Code Work <u>Build and Release</u>
Copy Files to: \$(build.artifactstagingdirectory)\projectartif	✓ (型) Documentation	
Publish Artifact: drop	V 🖻 Imaging Service	
	V 🖻 Marketing Site	
	V 🖻 Platform Administration	

## 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.

	Cont	tracts											
a	* U	Contract d	letails									× 14	Details
aurocaro.	o Te	Contract ID:	185-024									3	Dears
□ _	e Te	Involce Date:										3	Details
Accounts	+ 7t	Next Attempt	125 PM IS 2 months ag	10								3	Details
00	e Te										*	Details	
Part and	e Te									4	Dears		
=	e Tr	Test-Pres-3-Linge (Monthly) 82.80							2.00	Deals			
lings:	a Tr	Taluc	Attempts	10	Li	Starting	Dance		triding Salance		Actional Date	3	Details
	• 5	🛩 Paid	1	\$2	.00	\$0	.00		\$0.00		\$2.00	4	Details
	= M											4	Deals
	• B											OK 4	Details
	u Phe									-		4	Details
	e Star	ndard	\$1,299.00	8	45	4,190,820	308	35,808	30/30	100	316	Standard-2	Details
	• Con	npiete	S1,499.00	12	50	6,377,558	408	108,008	40.140	150	6.110	Standard-2	Details

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.

in the second		-				
	Sil records +					
8i	Time	Category	Activity	Description	Details	
	то 3 стания адо	Admission	subrenceson_Passed	Successful top in	Var exceededy regret in	lietar
	$\odot$ 52 minutes age	Custodiav	Rest Curtottan	for an intertaining account matter till in cases for ${\rm Matter}_{0}$ , ${\rm R}_{\rm rec}$		Detail
unte i con	0.14 minutes ago	Custodan	Curistian, Sleeping	Scheduled facts canadicis. Biosping for ${\mathbb Q}$ minutes	Taoto completed in 5,212 millipecands	Oetai
	😳 të manatos laga	Cuthdan	Cutation, Antonia Line (Cat., Kin.,	Reflecting account report of plants for Valuey Trade		Debal
	10 SH stituates ago	Carlolleri	Currotian, Raham Casha, Statur	Referring togs acl cache for 'Galary Tradets' at NEX		Oetai
	🗇 M minutes age	G, ehidan	Brox, Callolian	Error whetherg preparation and contro for Galaxy Trace		Drief
	10.14 minutes aga	Curtodian	Curses, Annah Cata, Ibas	Reflecting search-thosts api cache for 'Datary Tradest		Detail
	O 14 minution age	Custodien	Control on the Print Pri	References sensitive stations are called for Tailoup Track		Oviet
	1) Materia aga	CutoBall	Enorgicumplian	Einer terhenleng sondtr age cache for Yallong Tracter:		Detail
	0 M minutes agai	Cuntodian	Curbbar_SetenCeche_Dates	Partnering callegories api cache for "Dalacy Tradect"		Oetail

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.

1	Latest Transfers V 3 months ago \$0.66	O No scheduled barshers	Current Balance © Peedrop \$0.00	Credits >2 Total in circulation 0		
Dastraceted Accounts	Galaxy	ALL IN CATEGORY IN ACTIVITY IN ACT	COUNT BY PLATTING USED			
S Rains		Th.		2	TIT	
OS Paten	ing i	0 0 0	Inv	entory <b>Hawk</b>		
ta, Protec	PC Notes		Passe	ord Sign In		
	AL Press		Forgat you	r passwore? Reset (*		
				2		

# 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.