3rd Party Lock Logging Architectural Design Document

Document Control

Field	Value
Version:	1.0
Status:	ACCEPTED
Author(s)	@Umesh Kanyal
Reviewers:	@Longo, Jeffrey

Change History

Version	Date	Author	Description of Changes
1.0	4/30/2025	@Umesh Kanyal @Ritesh Dubey1 @Rahul Verma	Initial version

---

Table of Contents

Summary

This document focusses the Actions/SubActions and statuses that the KastleSDK will capture while configuring/interacting with 3rd party locks and provides a comprehensive overview of the KastleSDK system architecture, including its components, data flow, monitoring, performance requirements, and associated risks and mitigations. It outlines the integration of mobile applications with the KastleSDK for secure door operations, logging, and monitoring through Azure CosmosDB and Grafana.

System Context

Owner and Stakeholders

Role	Name(s)
Product Owners	@Burner, Todd @Reimer, Michael @De Leon, Cipriano
Tech Lead	@Longo, Jeffrey @Umesh Kanyal @Ritesh Dubey1
Team Lead	@Rahul Verma @Rahul Dubey @Deepak Yadav
SM	@Rahul Pandey

Objective

- Effective branch will be 4.1.x and module name will be **KastleLogger**
- Existing logging code logic need to be implemented by a separate module name **KastleLogger**
- Should have ticket for Allegion BLE, Dormakaba, HID and Allegion NFC monitoring changes.

Epic

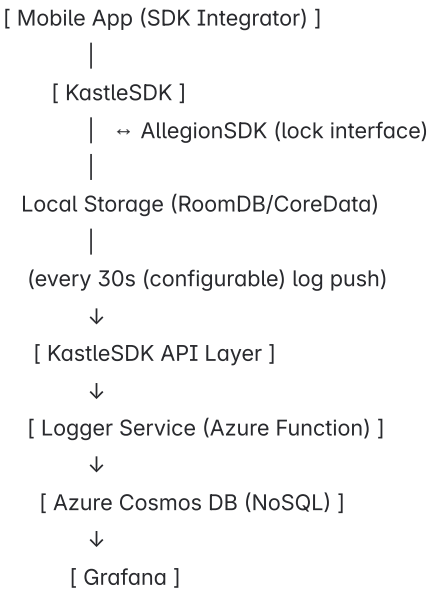
MFI-162: 3rd Party Lock Monitoring IN PROGRESS

Functional Requirement

Requirement	User Story	Importance	Acceptance Criteria	Jira Stories	Notes
				MOB-35119: | | FR || SDK || 3rd party lock logging || TO DO	

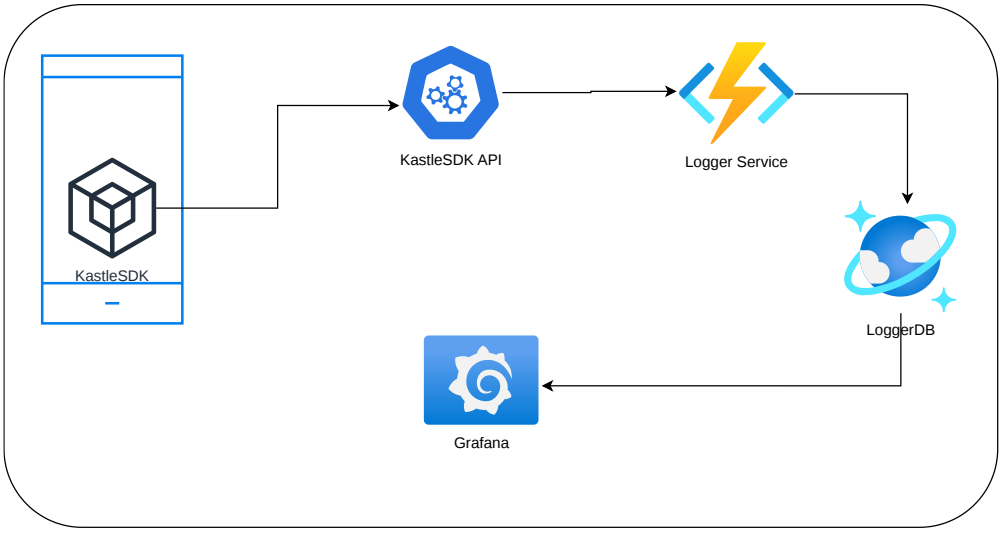
System Architecture

High-Level Architecture

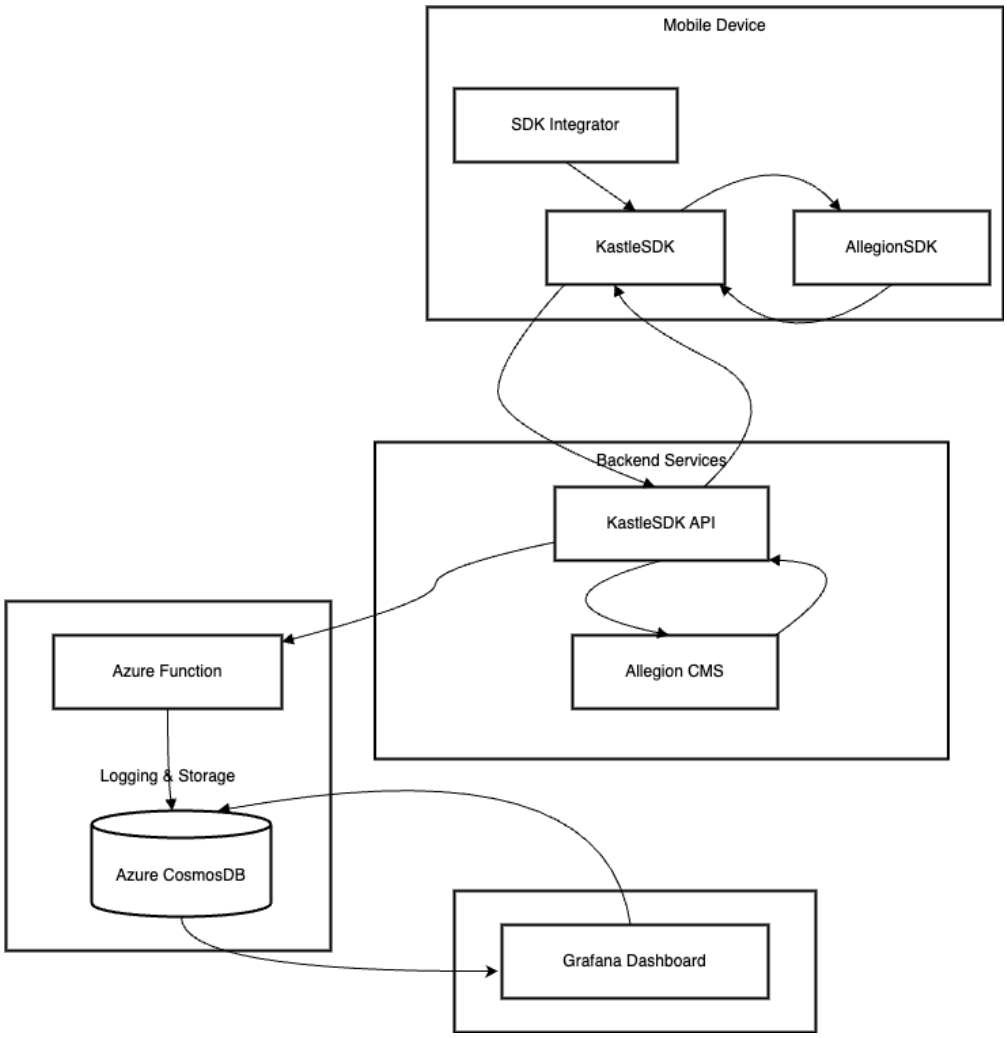


Component Model

The below component diagram is a generic which shows how the logging data will flow.



Components and data flow



Component	Role
SDK Integrator	End user application triggering the flow
KastleSDK	Core orchestrator — initiates commands and saves status
KastleSDK API	Backend logic + external communication handling

AllegionSDK	Lock/device control SDK
AllegionCMS	Credential/token provider
Logger Service	Azure Function that processes and forwards logs
Azure CosmosDB	Central time-series log storage
Grafana	Visualization, dashboard, and alerting tool

Data Architecture

Actions

Below are the Actions/SubActions (respective to each 3rd party) that would be required to tack the events and their states for different 3rd party SDK/Lock partners. It will help to identify at which stage the process failed (if any).

AllegionSDK

Action	SubAction	Module	Statuses	Description	Priority for Implementation
Authenticate		Allegion	Start,Retry,Success,Fail	In order to connect with Allegion CMS APIs, need to first get an authentication token	P1
DeleteCredential	cardformatid	Allegion	Start,Retry,Success,Fail	Before pushing the new credentials to Allegion CMS, delete the existing one for a cardholder	P1
CreateCredential	cardformatid	Allegion	Start,Retry,Success,Fail	Push Allegion credentials to AllegionCMS	P1
InitSDK		Allegion	Start,Retry,Success,Fail	Initialize the Allegion SDK	P1
DeviceEnrollment		Allegion	Start,Retry,Success,Fail	Device enrollment with AllegionSDK	P1
PullAccessRights		Allegion	Start,Retry,Success,Fail	Pull Access rights from AllegionSDK	P1
PullPayload		Allegion	Start,Retry,Success,Fail	Pull Payload from AllegionSDK	P1
WriteSecurityToken		Allegion	Start,Retry,Success,Fail	In order to connect with Allegion lock, the SDK needs to be authenticated by lock	P1
TouringOperation	Config, Schedule, PartialDb, FullDb, Audits	Allegion	Start,Retry,Success,Fail	Writing Config to lock	P1

Unlock	SDK	Allegion	Start,Retry,Success,Fail	Allegion Unlocking	P1

Allegion ServerCommand

Action	SubAction	Module	Statuses	Description	Priority
GetLockInfo		Allegion-ServerCommand	Start,Retry,Success,Fail	Fetches Server command details from API to unlock the door	P2
GetSiteInfo		Allegion-ServerCommand	Start,Retry,Success,Fail	Fetches secured site information	P2
CreateSecurityToken		Allegion-ServerCommand	Start,Retry,Success,Fail	Create an authentication command	P2
CreateServerCommand	currentSequence,reset,sequence_0	Allegion-ServerCommand	Start,Retry,Success,Fail	Create a currentSequence/ reset/ sequence_0 command, these commands are used to write current with current sequence, if it fails then app writes the reset command and then pushes with sequence_0	P2
WriteSecurityToken		Allegion-ServerCommand	Start,Retry,Success,Fail	Writing the Authentication command	P2
Unlock	ServerCommand_currentSeq, ServerCommand_reset, ServerCommand_seq_0	Allegion-ServerCommand	Start,Retry,Success,Fail	Write a currentSequence command if it fails then app writes the reset command and then pushes with sequence_0.	P2

HID

Event	SubEvent	Module	Status	Message	Priority
Authenticate		HID	Start,Retry,Success, Fail	OrigoMobileKeysApi provides token for further communication with APIs	P3
Initialization	EndPointSetup, RedeemToken	HID	Start,Retry,Success, Fail	Call to setup EndPoint with the token provided by Origo registration	P3
ProvisionWallet		HID	Start,Retry,Success, Fail	Provisioning Wallet	P3

Authorization		HID	Start,Retry,Success, Fail	Authorization	P3
CreatePass		HID	Start,Retry,Success, Fail	Create Pass	P3

Dormakaba

Action	SubAction	Module	Status	Message	**Priority**
Authenticate		Dormakaba	Start,Retry,Success, Fail	In case, SDK did not receive dormakaba token on registration, then call CreateDormakabaToken API	P1
InitSDK		Dormakaba	Start,Retry,Success, Fail	Dormakaba SDK initialization	P1
DeviceEnrollment		Dormakaba	Start,Retry,Success, Fail	Initiate registration with Dormakaba backend through Doramakaba SDK to identify the method to validate the token	P1
BackendRegistration Validation		Dormakaba	Start,Retry,Success, Fail	Registration With Backend using the Dormakaba token received on registration/ createdormakabatoken API	P1
CreateCredential	cardformat id	Dormakaba	Start, Retry, *Requested*, Success, Fail	Create Resident Key. Requested is when create resident returns a keyid. success is when synchronize returns available. fail is upon error or 15 minutes and no available key.	P1
CredentialStatus	activate, deactivate, **status_change**	Dormakaba	Start,Retry,Success, Fail **status_change states**: Available, Deployed, Remove_In_Progress, Removed	Dormakaba SDK pushes Neon file status	P1
Unlock		Dormakaba	Success, Fail	Reader unlocking callback	P1

Kastle

Action	**SubAction**	**Module**	**Status**	**Message/Description**	**Priority for implementation**

LogOverflow		Kastle	Occured	to be sent when overflow has occurred	P1
CreateCredential	cardformatid	KastleBLE	Start,Retry,Success, Fail		P3
Unlock	cardformatid	KastleBLE	Start,Retry,Success, Fail	When we actually interact with lock/reader.	P3

If any Action internally performs many subActions then logging should be in below format:

- **Action** without SubAction - "**Start**"
  - All the SubActions with their States i.e ("Start", "Success/Fail")
- **Action** without SubAction - "**Success/Fail**" depending upon state of SubActions.

example for Touring by PM

- TouringOperation Start
  - Config Start
  - Config Success/Fail
  - Schedule Start
  - Schedule Success/Fail
  - FullDb Start
  - FullDb Success/Fail
- TouringOperation Success/Fail

"Retry" status should be added in case the first or consecutive attempts fails, app should write "Fail" status if the last attempt also fails.

The same data model will be used for logging the details for different vendors like Allegion, HID, Dormakaba, though the Actions, SubActions would be different.

Data Model

Below is the data model that we are currently using to log all the actions and corresponding statuses.

DataModel

```
{
    "_id" : ObjectId("63e948cdf667f6f9284de997"),
    "SessionId" : "769248e7-089f-49a3-a456-5ad9dfc01fe1",
    "Action" : "TouringOperation",
    "SubAction":"Config",
    "Module" : "Allegion",
    "Message" : "TouringOperation-Config Start",
    "CardHolderGUID" : "1c992f0a-26b6-42fe-9e21-86cb939e57f7",
    "InstID" : "123456", //cardholder instid
```

```
    "EnterpriseID" : "123456",
    "BuildingNumber" : "ccbbbb",
    "DeviceTime" : "2025-04-26T19:00:00.000-07:00", //ISO 8601 format - local time with zone information from phone
    "Status" : "Start",
    "CardExternalNumber" : "40452-38844",
    "ApplicationIdentifier" : "com.KastleSystems.Presence",
    "AppVersion":"7.2",
    "SDKVersion":"4.1.9",
    "DeviceManufacturer":"Apple",
    "DeviceModel":"iphone12,0",
    "DeviceOS":"18.4",
    "LastUpdatedTime" : {
        "$date" : 1676232909000
    }
}
```

Data Model description:

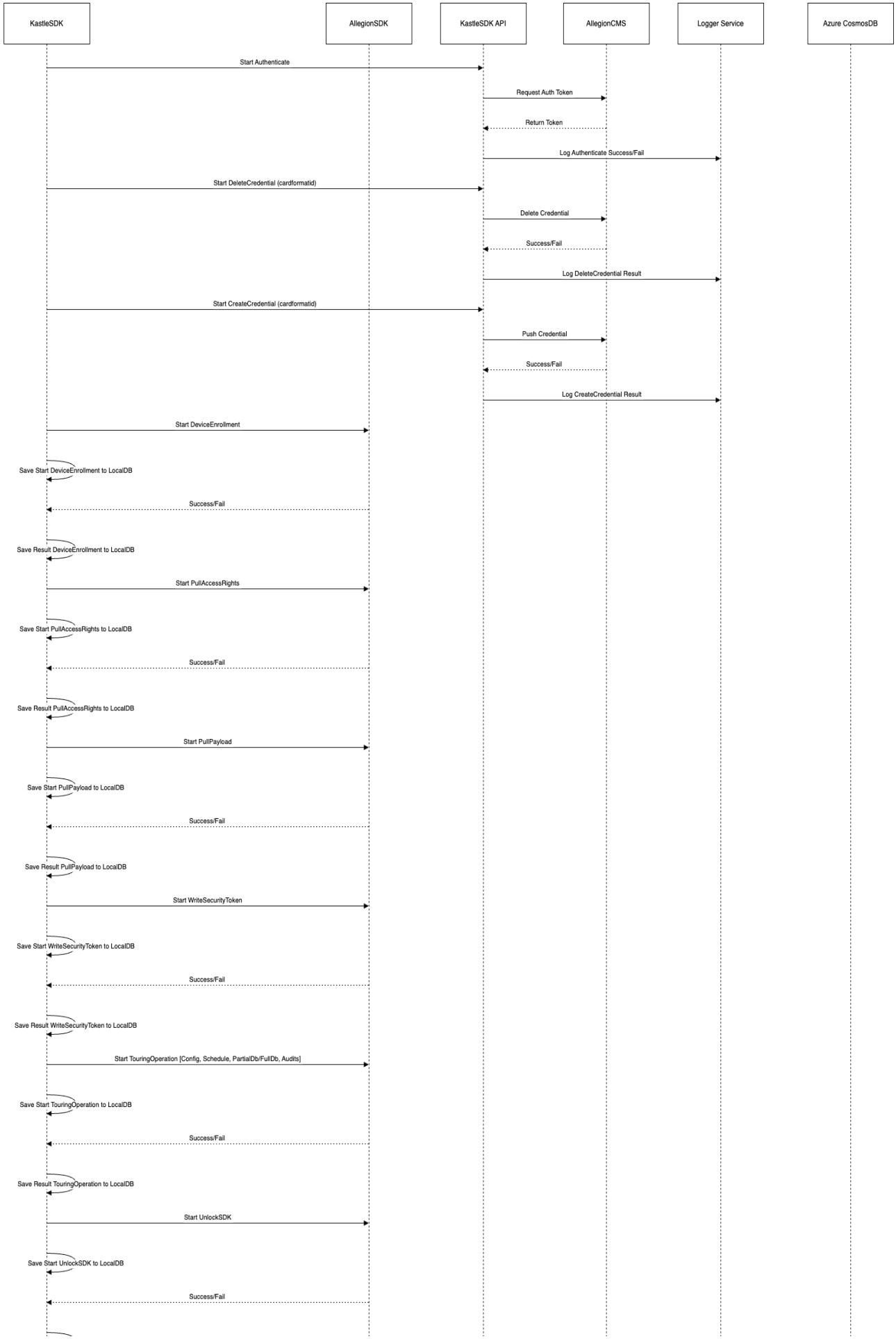
Property	Description	Example Value
`_id`	Unique identifier for the document, typically generated by the database.	ObjectId("63e948cdf667f6f9284de997")
`SessionId`	Unique identifier for the session in which the Action occurred. **Any action that has multiple subActions, all the subActions should also have the same SessionId**	"769248e7-089f-49a3-a456-5ad9dfc01fe1"
`Action`	Main operation being logged, representing a high-level task or command.	"TouringOperation"
`SubAction`	Specific subtask or operation under the main action.	"Config"
`Module`	what module	“Allegion”
`Message`	Descriptive message providing details about the action or its current state.	"TouringOperation-Config Start"
`CardHolderGUID`	Unique identifier for the cardholder associated with the operation.	"1c992f0a-26b6-42fe-9e21-86cb939e57f7"
`InstId`	Institution to which the cardholder belongs.	"300042426"
`EnterpriseId`	EntepriseId to which the cardholder belongs.	"30000001"
`BuildingNumber`	Building number “ccbbbb“ to which the cardholder belongs.	"DC0557"

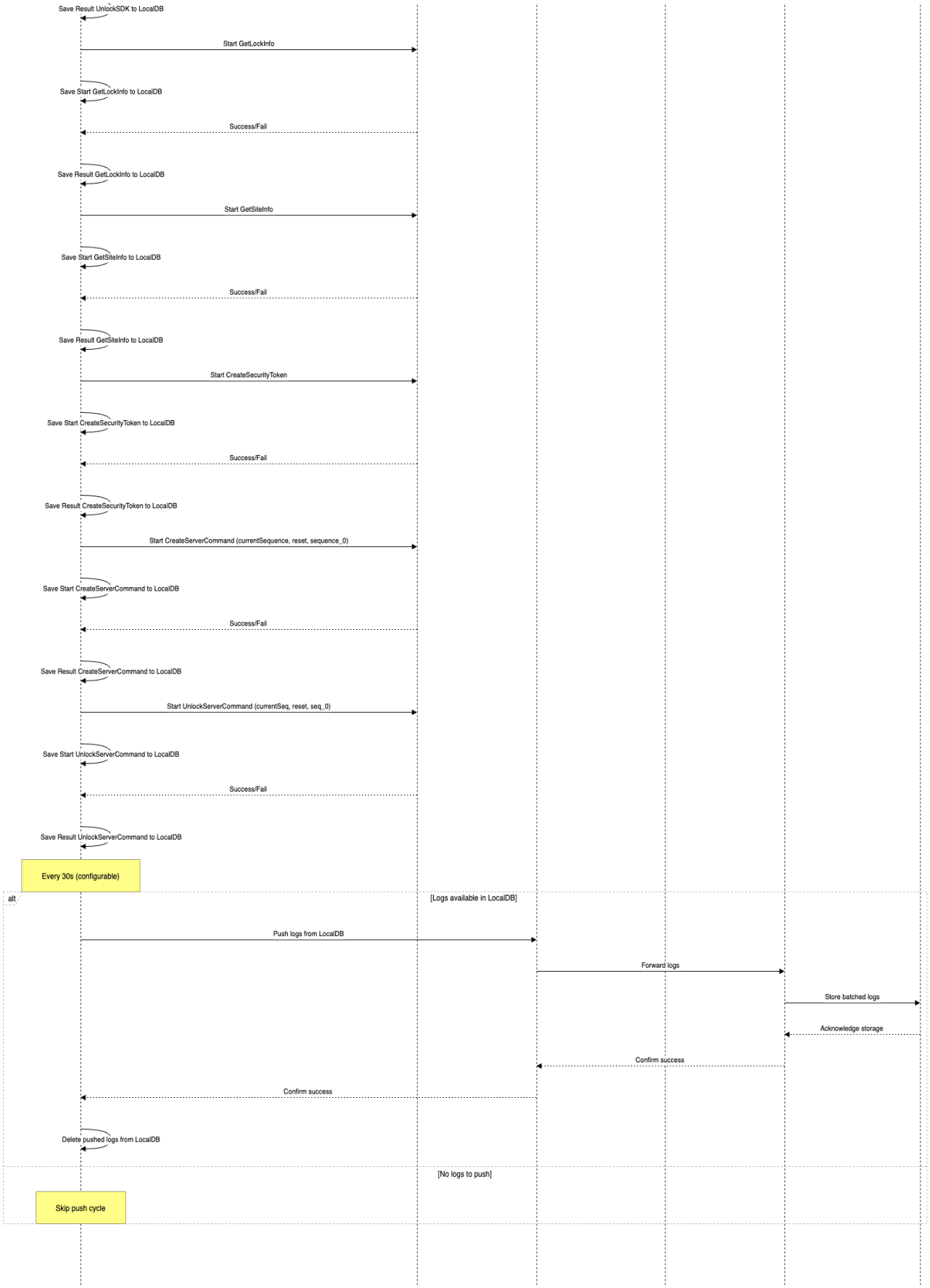
`DeviceTime`	Timestamp indicating when the action occurred on the device, in ISO 8601 format YYYY-MM-DDTHH:mm:ss.sssZ	"2025-04-26T19:00:00.000-07:00"
`Status`	Current status of the action (e.g., Started, Success, Fail).	"Start"
`CardExternalNumber`	External or human-readable identifier for the cardholder's card.	"40452-38844"
`ApplicationIdentifier`	Identifier for the application involved in the operation.	"com.KastleSystems.Presence"
`AppVersion`	Version of the application involved in the operation.	"7.2"
`SDKVersion`	Version of the SDK used for the operation.	"4.1.9"
`DeviceManufacturer`	Manufacturer of the device where the operation was performed.	"Apple"
`DeviceModel`	Model of the device where the operation was performed.	"iphone12,0"
`DeviceOS`	Operating system version of the device.	"18.4"
`LastUpdatedTime`	Timestamp (in milliseconds since epoch) indicating when the document was last updated. This is auto generated.	`1676232909000`

Data Flow

This sequence diagram outlines the end-to-end flow for integrating **KastleSDK** with a mobile or client application to perform secure door operations (e.g., registration, configuration, and unlock commands), while logging all operations via **Azure Cosmos DB**.

AllegionSDK


KastleSDK
AllegionSDK
KastleSDK API
AllegionCMS
Logger Service
Azure CosmosDB
Start Authenticate
Request Auth Token
Return Token
Log Authenticate Success/Fail
Start DeleteCredential (cardformatid)
Delete Credential
Success/Fail
Log DeleteCredential Result
Start CreateCredential (cardformatid)
Push Credential
Success/Fail
Log CreateCredential Result
Start DeviceEnrollment
Save Start DeviceEnrollment to LocalDB
Success/Fail
Save Result DeviceEnrollment to LocalDB
Start PullAccessRights
Save Start PullAccessRights to LocalDB
Success/Fail
Save Result PullAccessRights to LocalDB
Start PullPayload
Save Start PullPayload to LocalDB
Success/Fail
Save Result PullPayload to LocalDB
Start WriteSecurityToken
Save Start WriteSecurityToken to LocalDB
Success/Fail
Save Result WriteSecurityToken to LocalDB
Start TouringOperation [Config, Schedule, PartialDb/FullDb, Audits]
Save Start TouringOperation to LocalDB
Success/Fail
Save Result TouringOperation to LocalDB
Start UnlockSDK
Save Start UnlockSDK to LocalDB
Success/Fail

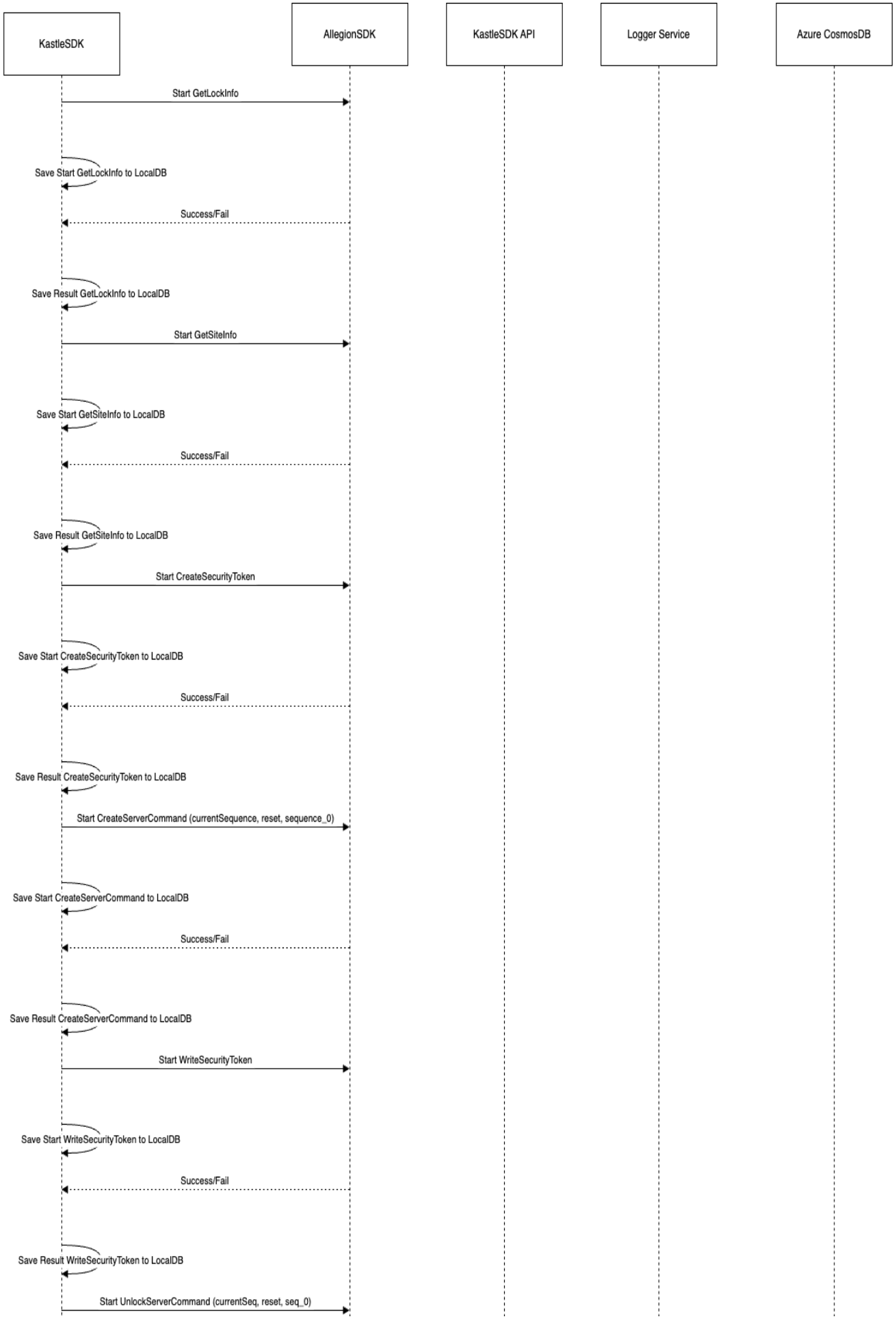


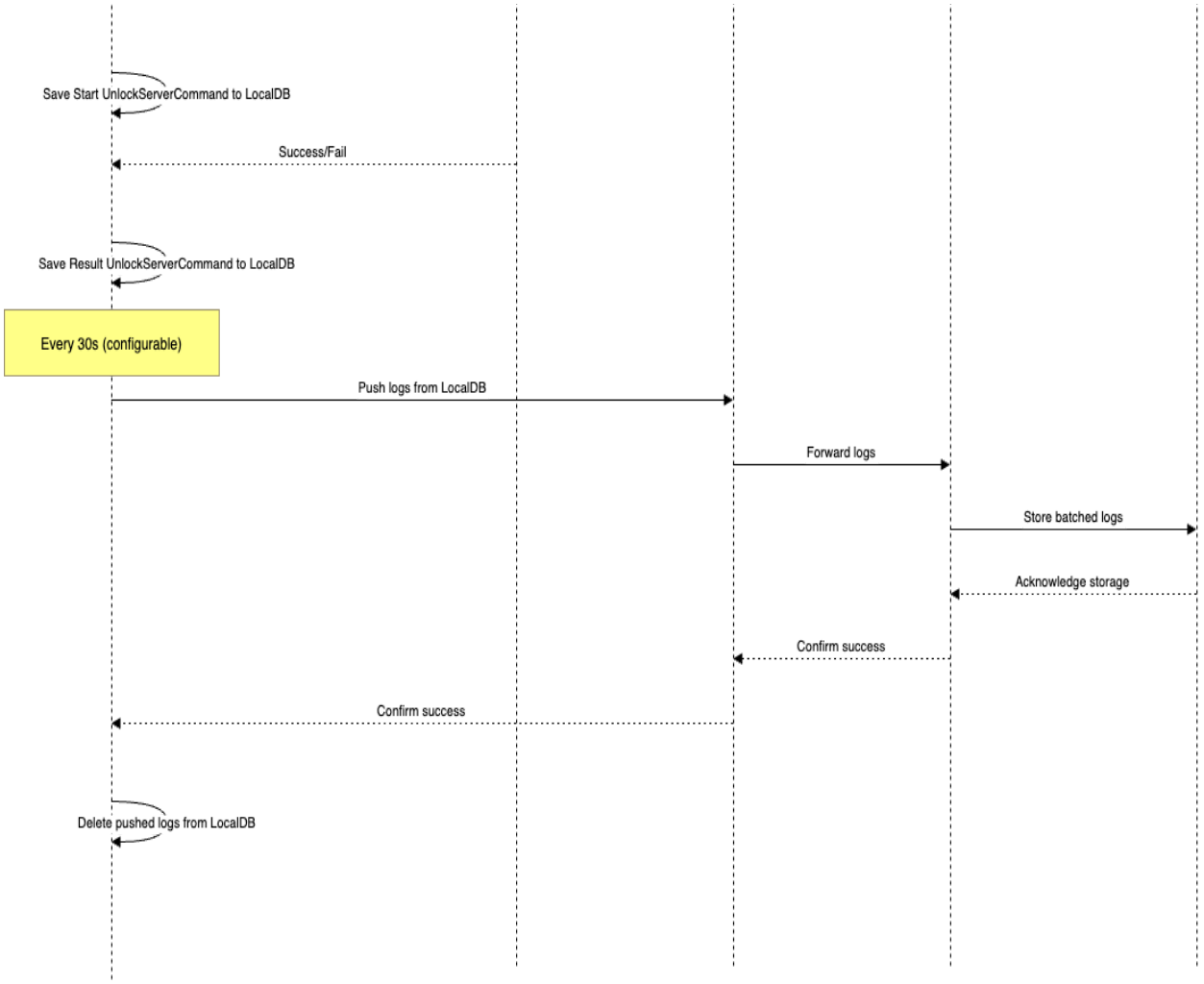

```
sequenceDiagram
    participant KastleSDK
    participant AllegionSDK
```

```
    participant KastleSDK API
    participant AllegionCMS
    participant Logger Service
    participant Azure CosmosDB

    %% Authenticate
    KastleSDK->>KastleSDK API: Start Authenticate
    KastleSDK API->>AllegionCMS: Request Auth Token
    AllegionCMS-->>KastleSDK API: Return Token
    KastleSDK API->>Logger Service: Log Authenticate Success/Fail

    %% DeleteCredential
    KastleSDK->>KastleSDK API: Start DeleteCredential (cardformatid)
    KastleSDK API->>AllegionCMS: Delete Credential
    AllegionCMS-->>KastleSDK API: Success/Fail
    KastleSDK API->>Logger Service: Log DeleteCredential Result

    %% CreateCredential
    KastleSDK->>KastleSDK API: Start CreateCredential (cardformatid)
    KastleSDK API->>AllegionCMS: Push Credential
    AllegionCMS-->>KastleSDK API: Success/Fail
    KastleSDK API->>Logger Service: Log CreateCredential Result

    %% DeviceEnrollment
    KastleSDK->>AllegionSDK: Start DeviceEnrollment
    KastleSDK->>KastleSDK: Save Start DeviceEnrollment to LocalDB
    AllegionSDK-->>KastleSDK: Success/Fail
    KastleSDK->>KastleSDK: Save Result DeviceEnrollment to LocalDB

    %% PullAccessRights
    KastleSDK->>AllegionSDK: Start PullAccessRights
    KastleSDK->>KastleSDK: Save Start PullAccessRights to LocalDB
    AllegionSDK-->>KastleSDK: Success/Fail
    KastleSDK->>KastleSDK: Save Result PullAccessRights to LocalDB

    %% PullPayload
    KastleSDK->>AllegionSDK: Start PullPayload
    KastleSDK->>KastleSDK: Save Start PullPayload to LocalDB
    AllegionSDK-->>KastleSDK: Success/Fail
    KastleSDK->>KastleSDK: Save Result PullPayload to LocalDB

    %% WriteSecurityToken
    KastleSDK->>AllegionSDK: Start WriteSecurityToken
    KastleSDK->>KastleSDK: Save Start WriteSecurityToken to LocalDB
    AllegionSDK-->>KastleSDK: Success/Fail
    KastleSDK->>KastleSDK: Save Result WriteSecurityToken to LocalDB

    %% TouringOperation
    KastleSDK->>AllegionSDK: Start TouringOperation [Config, Schedule, PartialDb/FullDb, Audits]
    KastleSDK->>KastleSDK: Save Start TouringOperation to LocalDB
    AllegionSDK-->>KastleSDK: Success/Fail
    KastleSDK->>KastleSDK: Save Result TouringOperation to LocalDB

    %% UnlockSDK
    KastleSDK->>AllegionSDK: Start UnlockSDK
    KastleSDK->>KastleSDK: Save Start UnlockSDK to LocalDB
    AllegionSDK-->>KastleSDK: Success/Fail
    KastleSDK->>KastleSDK: Save Result UnlockSDK to LocalDB
```

```
62
63      %%
64
65      %% Periodic Sync
66      Note over KastleSDK: Every 30s (configurable)
67      alt every 30s (configurable)
68          loop Sync logs in batches of 100 records
69              KastleSDK->>KastleSDK API: Push logs from LocalDB
70              KastleSDK API->>Logger Service: Forward logs
71              Logger Service->>Azure CosmosDB: Store batched logs
72              Azure CosmosDB-->>Logger Service: Acknowledge storage
73              Logger Service-->>KastleSDK API: Confirm success
74              KastleSDK API-->>KastleSDK: Confirm success
75              KastleSDK->>KastleSDK: Delete pushed logs from LocalDB
76          end
77      end
78
```

Allegion ServerCommand

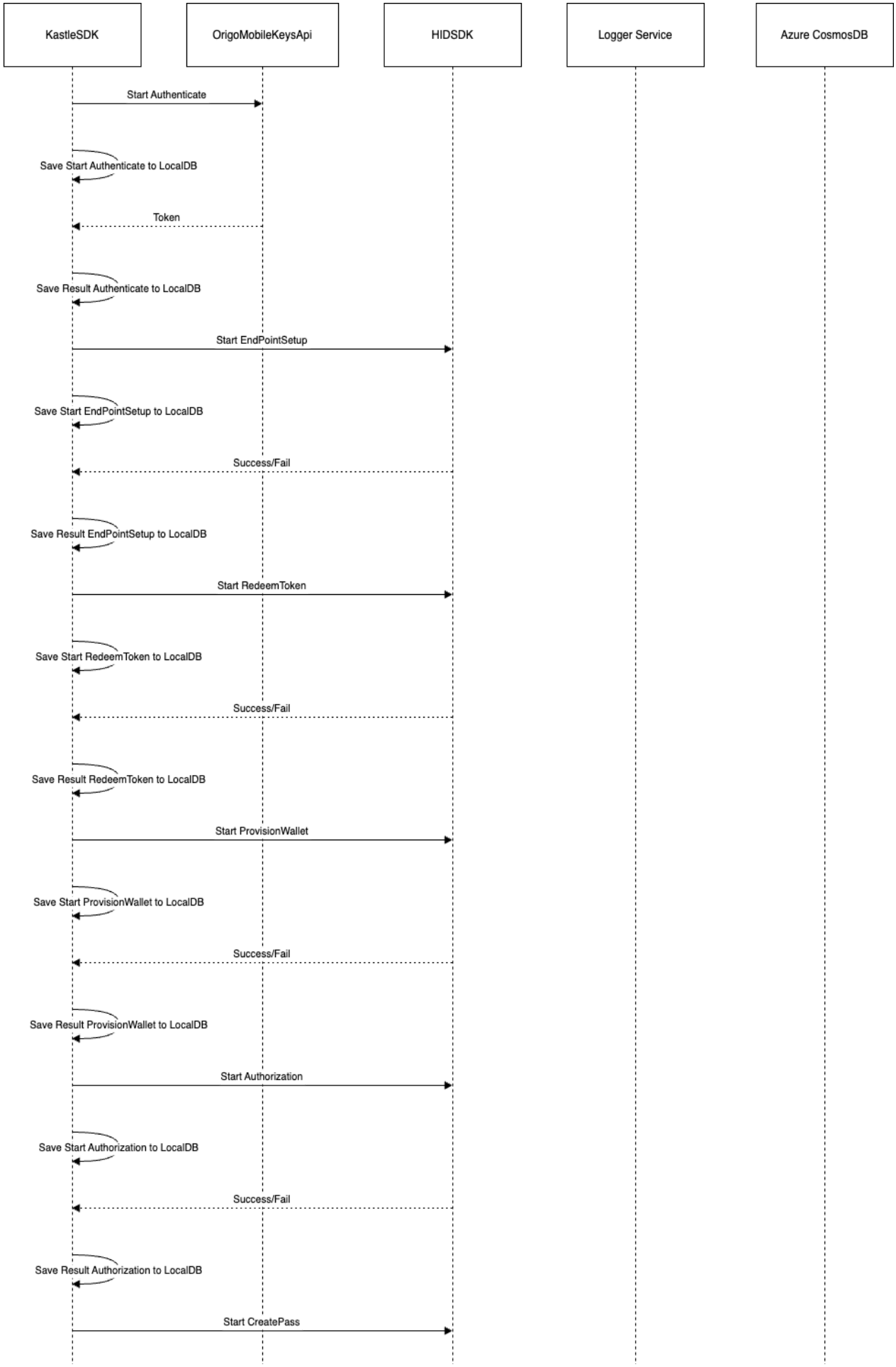

KastleSDK
AllegionSDK
KastleSDK API
Logger Service
Azure CosmosDB
Start GetLockInfo
Save Start GetLockInfo to LocalDB
Success/Fail
Save Result GetLockInfo to LocalDB
Start GetSiteInfo
Save Start GetSiteInfo to LocalDB
Success/Fail
Save Result GetSiteInfo to LocalDB
Start CreateSecurityToken
Save Start CreateSecurityToken to LocalDB
Success/Fail
Save Result CreateSecurityToken to LocalDB
Start CreateServerCommand (currentSequence, reset, sequence_0)
Save Start CreateServerCommand to LocalDB
Success/Fail
Save Result CreateServerCommand to LocalDB
Start WriteSecurityToken
Save Start WriteSecurityToken to LocalDB
Success/Fail
Save Result WriteSecurityToken to LocalDB
Start UnlockServerCommand (currentSeq, reset, seq_0)

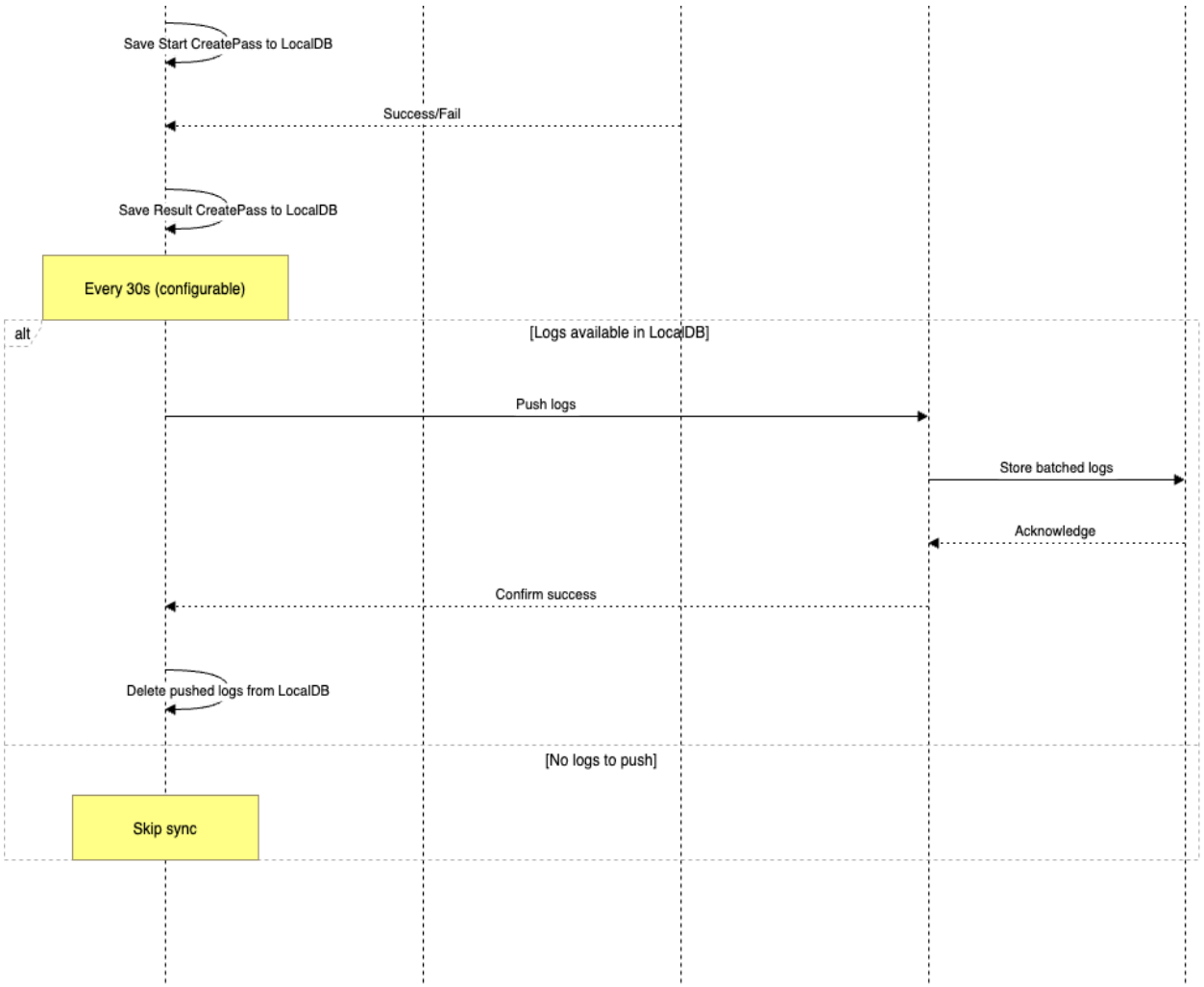



```
sequenceDiagram
    participant KastleSDK
    participant AllegionSDK
    participant KastleSDK API
    participant Logger Service
    participant Azure CosmosDB

    %% GetLockInfo
    KastleSDK->>AllegionSDK: Start GetLockInfo
    KastleSDK->>KastleSDK: Save Start GetLockInfo to LocalDB
    AllegionSDK-->>KastleSDK: Success/Fail
    KastleSDK->>KastleSDK: Save Result GetLockInfo to LocalDB

    %% GetSiteInfo
    KastleSDK->>AllegionSDK: Start GetSiteInfo
    KastleSDK->>KastleSDK: Save Start GetSiteInfo to LocalDB
    AllegionSDK-->>KastleSDK: Success/Fail
    KastleSDK->>KastleSDK: Save Result GetSiteInfo to LocalDB

    %% CreateSecurityToken
    KastleSDK->>AllegionSDK: Start CreateSecurityToken
    KastleSDK->>KastleSDK: Save Start CreateSecurityToken to LocalDB
    AllegionSDK-->>KastleSDK: Success/Fail
    KastleSDK->>KastleSDK: Save Result CreateSecurityToken to LocalDB

    %% CreateServerCommand
```

```
    KastleSDK->>AllegionSDK: Start CreateServerCommand (currentSequence, reset, sequence_0)
    KastleSDK->>KastleSDK: Save Start CreateServerCommand to LocalDB
    AllegionSDK-->>KastleSDK: Success/Fail
    KastleSDK->>KastleSDK: Save Result CreateServerCommand to LocalDB

    %% WriteSecurityToken
    KastleSDK->>AllegionSDK: Start WriteSecurityToken
    KastleSDK->>KastleSDK: Save Start WriteSecurityToken to LocalDB
    AllegionSDK-->>KastleSDK: Success/Fail
    KastleSDK->>KastleSDK: Save Result WriteSecurityToken to LocalDB

    %% UnlockServerCommand
    KastleSDK->>AllegionSDK: Start UnlockServerCommand (currentSeq, reset, seq_0)
    KastleSDK->>KastleSDK: Save Start UnlockServerCommand to LocalDB
    AllegionSDK-->>KastleSDK: Success/Fail
    KastleSDK->>KastleSDK: Save Result UnlockServerCommand to LocalDB

    %%

    %% Periodic Sync
     Note over KastleSDK: Every 30s (configurable)
    alt every 30s (configurable)
        loop Sync logs in batches of 100 records
            KastleSDK->>KastleSDK API: Push logs from LocalDB
            KastleSDK API->>Logger Service: Forward logs
            Logger Service->>Azure CosmosDB: Store batched logs
            Azure CosmosDB-->>Logger Service: Acknowledge storage
            Logger Service-->>KastleSDK API: Confirm success
            KastleSDK API-->>KastleSDK: Confirm success
            KastleSDK->>KastleSDK: Delete pushed logs from LocalDB
        end
    end
```

HID

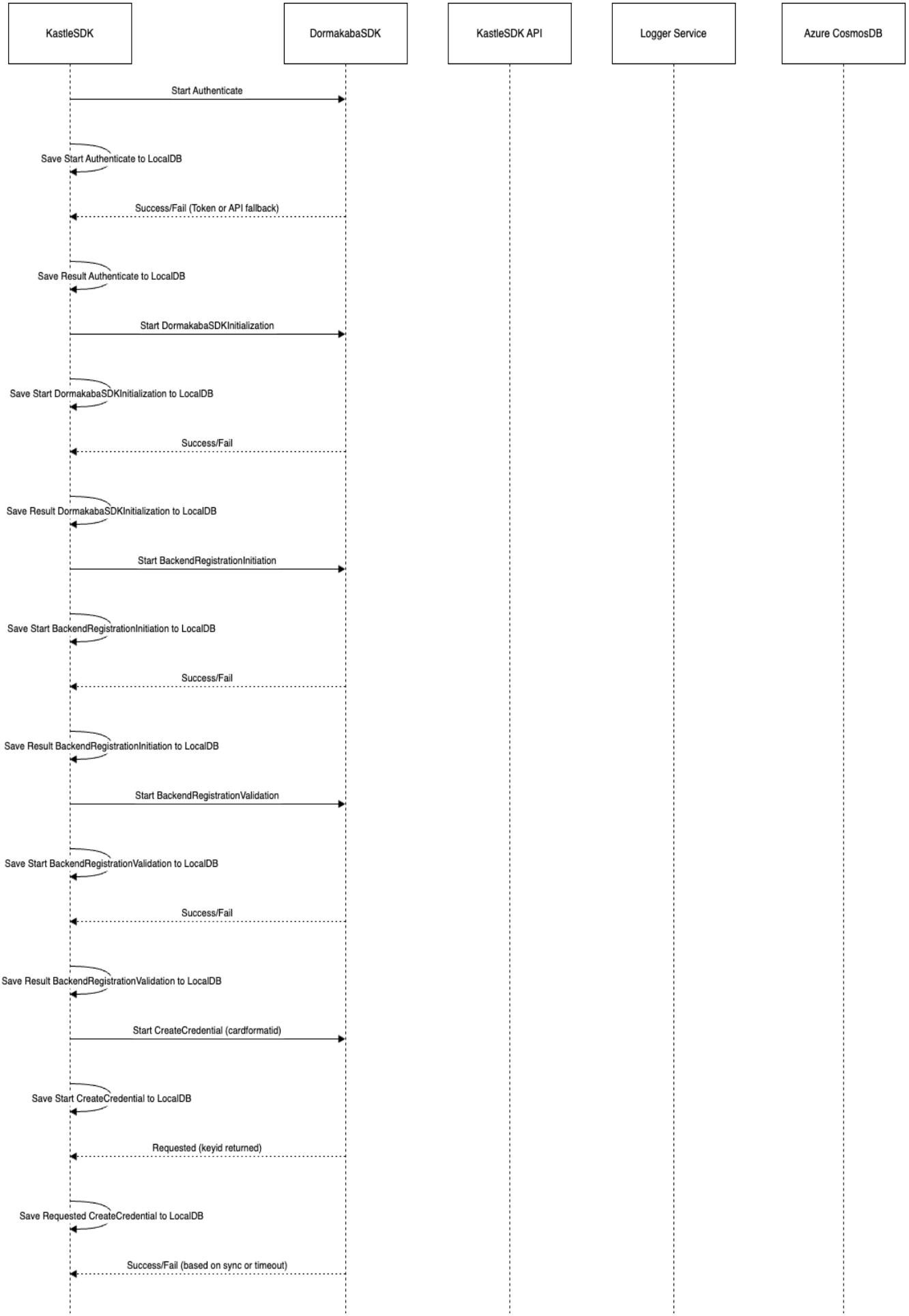

KastleSDK
OrigoMobileKeysApi
HIDSDK
Logger Service
Azure CosmosDB
Start Authenticate
Save Start Authenticate to LocalDB
Token
Save Result Authenticate to LocalDB
Start EndPointSetup
Save Start EndPointSetup to LocalDB
Success/Fail
Save Result EndPointSetup to LocalDB
Start RedeemToken
Save Start RedeemToken to LocalDB
Success/Fail
Save Result RedeemToken to LocalDB
Start ProvisionWallet
Save Start ProvisionWallet to LocalDB
Success/Fail
Save Result ProvisionWallet to LocalDB
Start Authorization
Save Start Authorization to LocalDB
Success/Fail
Save Result Authorization to LocalDB
Start CreatePass

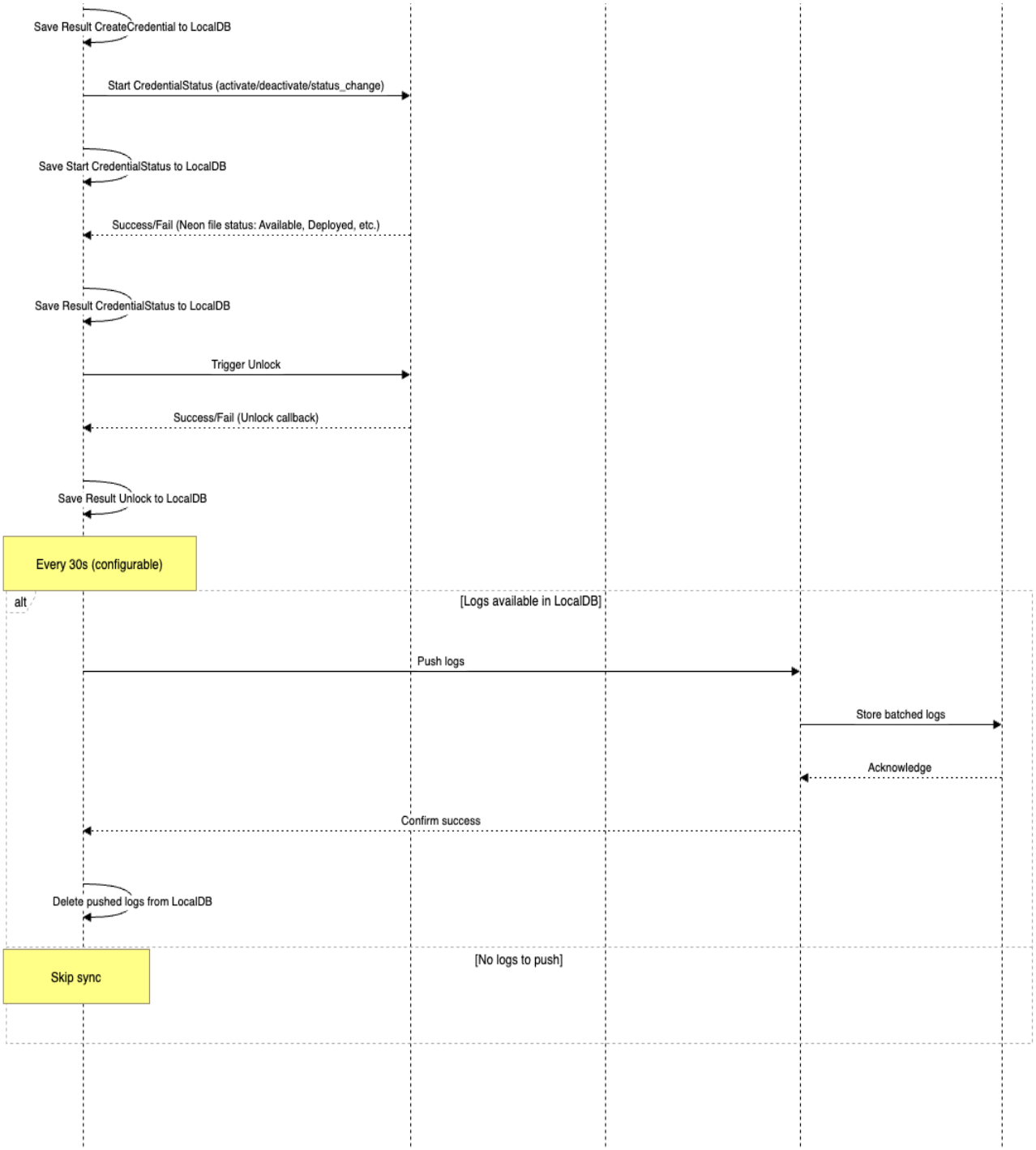



```
sequenceDiagram
    participant KastleSDK
    participant OrigoMobileKeysApi
    participant HIDSDK
    participant Logger Service
    participant Azure CosmosDB

    %% Authenticate
    KastleSDK->>OrigoMobileKeysApi: Start Authenticate
    KastleSDK->>KastleSDK: Save Start Authenticate to LocalDB
    OrigoMobileKeysApi-->>KastleSDK: Token
    KastleSDK->>KastleSDK: Save Result Authenticate to LocalDB

    %% Initialization - EndPointSetup
    KastleSDK->>HIDSDK: Start EndPointSetup
    KastleSDK->>KastleSDK: Save Start EndPointSetup to LocalDB
    HIDSDK-->>KastleSDK: Success/Fail
    KastleSDK->>KastleSDK: Save Result EndPointSetup to LocalDB

    %% Initialization - RedeemToken
    KastleSDK->>HIDSDK: Start RedeemToken
    KastleSDK->>KastleSDK: Save Start RedeemToken to LocalDB
    HIDSDK-->>KastleSDK: Success/Fail
    KastleSDK->>KastleSDK: Save Result RedeemToken to LocalDB

    %% ProvisionWallet
```

```
    KastleSDK->>HIDSDK: Start ProvisionWallet
    KastleSDK->>KastleSDK: Save Start ProvisionWallet to LocalDB
    HIDSDK-->>KastleSDK: Success/Fail
    KastleSDK->>KastleSDK: Save Result ProvisionWallet to LocalDB

    %% Authorization
    KastleSDK->>HIDSDK: Start Authorization
    KastleSDK->>KastleSDK: Save Start Authorization to LocalDB
    HIDSDK-->>KastleSDK: Success/Fail
    KastleSDK->>KastleSDK: Save Result Authorization to LocalDB

    %% CreatePass
    KastleSDK->>HIDSDK: Start CreatePass
    KastleSDK->>KastleSDK: Save Start CreatePass to LocalDB
    HIDSDK-->>KastleSDK: Success/Fail
    KastleSDK->>KastleSDK: Save Result CreatePass to LocalDB

  %%

    %% Periodic Sync
     Note over KastleSDK: Every 30s (configurable)
    alt every 30s (configurable)
        loop Sync logs in batches of 100 records

            KastleSDK->>KastleSDK API: Push logs from LocalDB
            KastleSDK API->>Logger Service: Forward logs
            Logger Service->>Azure CosmosDB: Store batched logs
            Azure CosmosDB-->>Logger Service: Acknowledge storage
            Logger Service-->>KastleSDK API: Confirm success
            KastleSDK API-->>KastleSDK: Confirm success
            KastleSDK->>KastleSDK: Delete pushed logs from LocalDB
        end
    end
```

Dormakaba


KastleSDK
DormakabaSDK
KastleSDK API
Logger Service
Azure CosmosDB
Start Authenticate
Save Start Authenticate to LocalDB
Success/Fail (Token or API fallback)
Save Result Authenticate to LocalDB
Start DormakabaSDKInitialization
Save Start DormakabaSDKInitialization to LocalDB
Success/Fail
Save Result DormakabaSDKInitialization to LocalDB
Start BackendRegistrationInitiation
Save Start BackendRegistrationInitiation to LocalDB
Success/Fail
Save Result BackendRegistrationInitiation to LocalDB
Start BackendRegistrationValidation
Save Start BackendRegistrationValidation to LocalDB
Success/Fail
Save Result BackendRegistrationValidation to LocalDB
Start CreateCredential (cardformatid)
Save Start CreateCredential to LocalDB
Requested (keyid returned)
Save Requested CreateCredential to LocalDB
Success/Fail (based on sync or timeout)




```
sequenceDiagram
    participant KastleSDK
    participant DormakabaSDK
    participant KastleSDK API
    participant Logger Service
    participant Azure CosmosDB

    %% Authenticate
    KastleSDK->>DormakabaSDK: Start Authenticate
    KastleSDK->>KastleSDK: Save Start Authenticate to LocalDB
    DormakabaSDK-->>KastleSDK: Success/Fail (Token or API fallback)
    KastleSDK->>KastleSDK: Save Result Authenticate to LocalDB

    %% Dormakaba SDK Initialization
```

```
    KastleSDK->>DormakabaSDK: Start DormakabaSDKInitialization
    KastleSDK->>KastleSDK: Save Start DormakabaSDKInitialization to LocalDB
    DormakabaSDK-->>KastleSDK: Success/Fail
    KastleSDK->>KastleSDK: Save Result DormakabaSDKInitialization to LocalDB

    %% BackendRegistrationInitiation
    KastleSDK->>DormakabaSDK: Start BackendRegistrationInitiation
    KastleSDK->>KastleSDK: Save Start BackendRegistrationInitiation to LocalDB
    DormakabaSDK-->>KastleSDK: Success/Fail
    KastleSDK->>KastleSDK: Save Result BackendRegistrationInitiation to LocalDB

    %% BackendRegistrationValidation
    KastleSDK->>DormakabaSDK: Start BackendRegistrationValidation
    KastleSDK->>KastleSDK: Save Start BackendRegistrationValidation to LocalDB
    DormakabaSDK-->>KastleSDK: Success/Fail
    KastleSDK->>KastleSDK: Save Result BackendRegistrationValidation to LocalDB

    %% CreateCredential
    KastleSDK->>DormakabaSDK: Start CreateCredential (cardformatid)
    KastleSDK->>KastleSDK: Save Start CreateCredential to LocalDB
    DormakabaSDK-->>KastleSDK: Requested (keyid returned)
    KastleSDK->>KastleSDK: Save Requested CreateCredential to LocalDB
    DormakabaSDK-->>KastleSDK: Success/Fail (based on sync or timeout)
    KastleSDK->>KastleSDK: Save Result CreateCredential to LocalDB

    %% CredentialStatus
    KastleSDK->>DormakabaSDK: Start CredentialStatus (activate/deactivate/status_change)
    KastleSDK->>KastleSDK: Save Start CredentialStatus to LocalDB
    DormakabaSDK-->>KastleSDK: Success/Fail (Neon file status: Available, Deployed, etc.)
    KastleSDK->>KastleSDK: Save Result CredentialStatus to LocalDB

    %% Unlock
    KastleSDK->>DormakabaSDK: Trigger Unlock
    DormakabaSDK-->>KastleSDK: Success/Fail (Unlock callback)
    KastleSDK->>KastleSDK: Save Result Unlock to LocalDB

  %%

    %% Periodic Sync
     Note over KastleSDK: Every 30s (configurable)
    alt every 30s (configurable)
        loop Sync logs in batches of 100 records
            KastleSDK->>KastleSDK API: Push logs from LocalDB
            KastleSDK API->>Logger Service: Forward logs
            Logger Service->>Azure CosmosDB: Store batched logs
            Azure CosmosDB-->>Logger Service: Acknowledge storage
            Logger Service-->>KastleSDK API: Confirm success
            KastleSDK API-->>KastleSDK: Confirm success
            KastleSDK->>KastleSDK: Delete pushed logs from LocalDB
        end
    end
```

Storage Layer

Database Systems:

1. Local Storage (within KastleSDK)
    a. iOS - CoreData

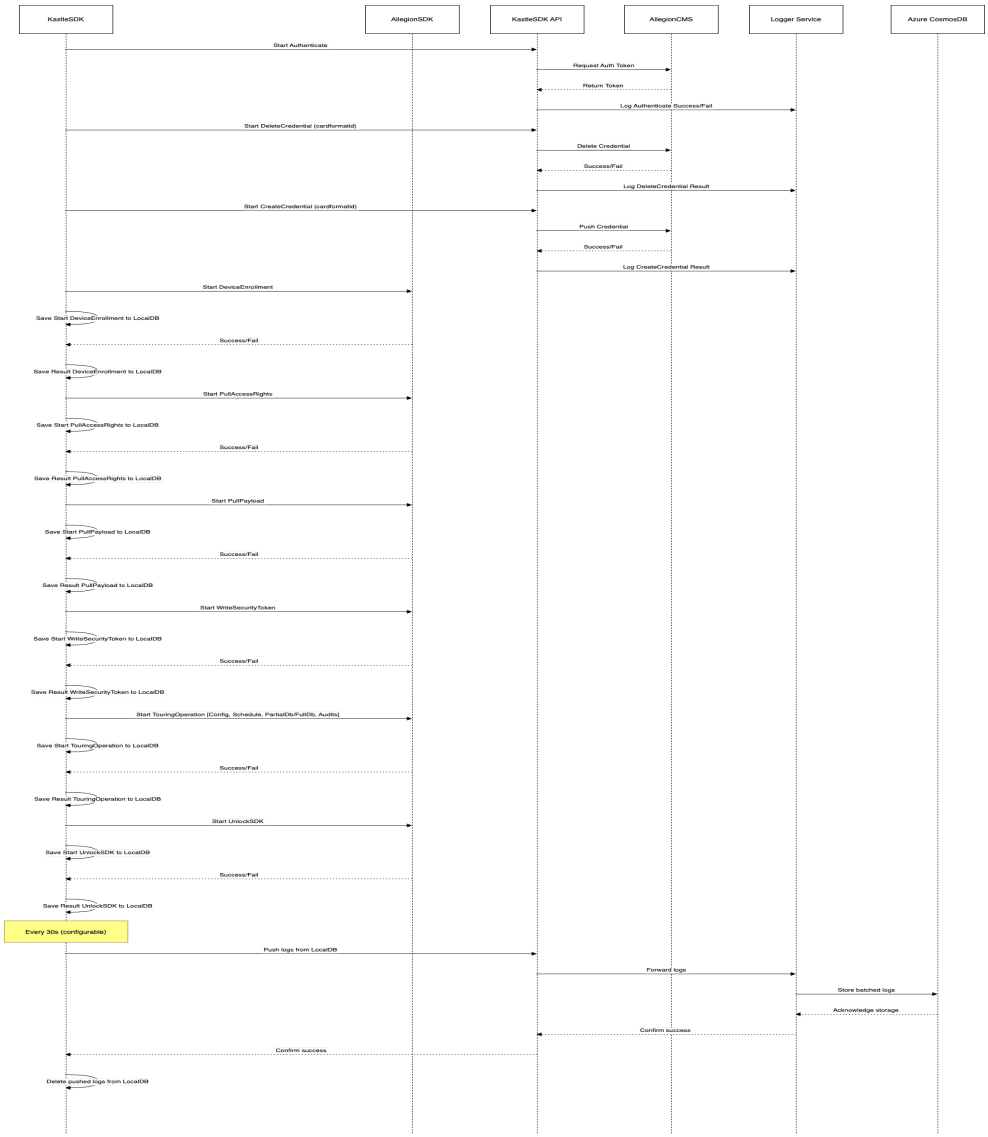
b. Android - RoomDB

2. Final Storage Layer - Azure Cosmos DB

Data Retention:

For Local storage, the maximum number of log messages to be queued shall be 1000. Once 1000 messages are queued, set flag indicating Log Overflow, and inject a log indicating log overflow next time we transmit logs up and then clear the flag once transmission is successful.

Transmit in batches of 100 until queue below batch amount.



	Local	**Server**
Retention Period:	Indefinite until sent to server	90 days
Archival Strategy:	Store and forward	statistics rollup

System Behavior

Monitoring

Will use **Grafana** to **monitor, alert, and analyze** every part of the device registration and door unlock flow handled via **KastleSDK**, backed by logs stored in **Azure Cosmos DB**.

AllegionSDK

Action	Sub-Action	Monitored Data Points	Grafana Panel	Trigger for Warning (email)	Trigger for Alerts (wake people up)
Authenticate	—	Auth token request success/failure rate, response time	Allegion > CMS Authentication Health	Average Failure rate > 2% or response time > 4s over rolling 10m	Average Failure rate > 5% or response time > 4s over rolling 10m
DeleteCredential	cardformatid	Deletion status, duration per request	Allegion > Credential Lifecycle	Average Failure rate > 2% over rolling 10m	Average Failure rate > 5% over rolling 10m
CreateCredential	cardformatid	Creation flow, credential push outcome	Allegion > Credential Lifecycle	Average Failure rate > 2% or response time > 4s over rolling 10m	Average Failure rate > 5% or response time > 4s over rolling 10m
DeviceEnrollment	—	Enrollment success/fail, SDK response time	Allegion > Device Enrollment	Average Failure rate > 2% or response time > 15s over rolling 10m	Average Failure rate > 5% or response time > 25s over rolling 10m
PullAccessRights	local,remote	Access rights pull the success rate and data size	Allegion > Rights Pull Health	Average Failure rate > 2% or response time > 15s over rolling 10m	Average Failure rate > 5% or response time > 25s over rolling 10m
PullPayload	cache, remote	Payload pull volume, completeness, and latency	Allegion > Payload Sync Status	Average Failure rate > 2% or response time > 15s over rolling 10m	Average Failure rate > 5% or response time > 25s over rolling 10m
WriteSecurityToken	—	Token write response, success/fail trend	Allegion > Lock Authentication	Average Failure rate > 1% over rolling 10m	Average Failure rate > 2% over rolling 10m
TouringOperation	Config, Schedule, PartialDb, FullDb, Audits	Config write latency, by operation type	Allegion > Touring Configuration Sync	Average Failure rate > 3% over rolling 10m	Average Failure rate > 5% over rolling 10m
UnlockSDK	—	Unlock request path, callback time	Allegion > Unlock Operation	Failure rate > 1% over rolling 10m	Failure rate > 2% over rolling 10m

No Data			Allegion Submodule Data	No actions logged by Allegion submodule for last 15m between 8am and 10pm ET	No actions logged by Allegion submodule for last 30m between 8am and 10pm ET

Allegion ServerCommand

Action	Sub-Action	Monitored Data Points	Grafana Panel	Trigger for Warnings (email)	Trigger for Alerts (wake people up)
GetLockInfo	—	API response latency, success/failure rate	Allegion > Lock Info API Health	Average Failure rate > 1% or average response time > 2s over rolling 10m	Average Failure rate > 3% or average response time > 5s over rolling 10m
GetSiteInfo	—	Site info API success/failure, data availability	Site Info Lookup	Average Failure rate > 1% or average response time > 2s over rolling 10m	Average Failure rate > 3% or average response time > 5s over rolling 10m
CreateSecurityToken	—	Command success, token generation time	Token Creation Commands	Average Failure rate > 1% over rolling 10m	Average Failure rate > 3% over rolling 10m
CreateServerCommand	currentSequence, reset, sequence_0	Command fallback path, retry attempts, success/failure patterns	Server Command Fallback Health	Average Failure rate > 1% over rolling 10m	Average Failure rate > 3% over rolling 10m
WriteSecurityToken	—	Write command result, retry occurrence	Allegion > Token Write Metrics	Average Failure rate > 1% over rolling 10m	Average Failure rate > 2% over rolling 10m
Unlock	ServerCommand_currentSeq, ServerCommand_reset, ServerCommand_seq_0	Execution path (currentSeq vs reset vs seq_0), timing, failure	Allegion > Unlock Command Path Analysis	Average Failure rate > 1% over rolling 10m	Average Failure rate > 2% over rolling 10m

HID

Action	Sub-Action	Monitored Data Points	Grafana Panel	Trigger for Warnings (email)	Trigger for Alerts (wake people up)
Authenticate	—	Token attempts, success/fail count	HID > Authentication Token Health	Average Failure rate > 1% over rolling 10m	Average Failure rate > 3% over rolling 10m

Initialization	EndPointSetup	SDK setup attempts and responses	HID > Endpoint Setup Stability	Average Failure rate > 1% over rolling 10m	Average Failure rate > 3% over rolling 10m
Initialization	RedeemToken	Redemption success/fail	HID > Token Redeem Metrics	Average Failure rate > 1% over rolling 10m	Average Failure rate > 3% over rolling 10m
ProvisionWallet	—	Wallet provisioning attempts and failures	HID > Wallet Provisioning	Average Failure rate > 1% over rolling 10m	Average Failure rate > 3% over rolling 10m
Authorization	—	Auth events, token match status	HID > Authorization Success Rate	Average Failure rate > 1% over rolling 10m	Average Failure rate > 3% over rolling 10m
CreatePass	—	Pass creation lifecycle (Requested, Success, Fail)	HID > Pass Creation Health	Average Failure rate > 1% over rolling 10m	Average Failure rate > 3% over rolling 10m

Dormakaba

Action	Sub-Action	Monitored Data Points	Grafana Panel	Trigger for Warnings (email)	Trigger for Alerts (wake people up)
Authenticate	—	Token status: received vs fallback API	Dormakaba > Token Acquisition Health	Average Failure rate > 2% across attempts over rolling 10m	Average Failure rate > 5% across attempts over rolling 10m
DormakabaSDKInitialization	—	SDK init success/fail, versions	Dormakaba > SDK Initialization	Average Init failure rate > 2% over rolling 10m	Average Init failure rate > 5% over rolling 10m
BackendRegistrationInitiation	—	Method resolution stats, backend status	Dormakaba > Registration Initiation	Average failure rate > 2% over rolling 10m	Average failure rate > 5% over rolling 10m
BackendRegistrationValidation	—	Token validation status with the backend	Dormakaba > Registration Validation	Average Validation failure rate > 2% over rolling 10m	Average Validation failure rate > 5% over rolling 10m
CreateCredential	cardformatid	Requested → Success flow, failures, key availability	Dormakaba > Credential Creation Funnel	No success or fail > 2% within 15 mins	No success or fail > 5% within 15 mins
CredentialStatus	activate, deactivate, status_change	Transition tracking, current vs previous state	Dormakaba > Credential Status Monitoring	Average Invalid/unknown states > 2% of transitions over rolling 10m	Average Invalid/unknown states > 5% of transitions over rolling 10m

Unlock	—	Unlock the callback outcome from a reader	Dormakaba > Unlock Success Rate	Average Unlock failure rate > 1% over rolling 10m	Average Unlock failure rate > 2% over rolling 10m

---

Cross-Cutting Concerns

Performance Architecture

Performance Requirements:

AllegionSDK

Action	Performance Metric	Requirement
Authenticate	Token response time	≤ 1200-1600 ms
	Failure rate	≤ 1%
DeleteCredential	Deletion latency	≤ 900-1200 ms
	Failure rate	≤ 1%
CreateCredential	Credential push latency	≤ 1600-2000 ms
	Failure rate	≤ 1%
DeviceEnrollment	Enrollment execution time	≤ 15 seconds
	Failure rate	≤ 1%
PullAccessRights	Response latency	≤ 1200-1600 ms
	Failure rate	≤ 1%
PullPayload	Pull time	≤ 1200-1600 ms
	Failure rate	≤ 1%
WriteSecurityToken	Authenticate with the lock	≤ 300
	Failure rate	≤ 1%
TouringOperation	Config write duration	≤ 5 seconds (any config type)
	Failure rate	≤ 5%
Unlock	Unlock callback response time	≤ 3 seconds iOS ≤ 5 seconds Android
	Unlock failure rate	≤ 1%

Allegion ServerCommand

Action	Performance Metric	Requirement

GetLockInfo	API response time	≤ 1200-1600 ms
GetSiteInfo	Site metadata availability	≤ 1200-1600 ms
CreateSecurityToken	Token generation response	≤ 200 ms
CreateServerCommand	Primary sequence write success time	≤ 100 ms
WriteSecurityToken	Token push to device	≤ 500 ms
Unlock	Total unlock execution (with fallbacks)	≤ 1500 - 1800 ms ≤ 3 seconds Android
All Actions	Failure rate threshold	≤ 2% generally; ≤ 2% for unlock or fallback-intensive actions
All Actions	Grafana panel ingestion delay	≤ 10 seconds from local log

HID

Action	Performance Metric	Requirement
Authenticate	Token issuance latency	≤ 1500-1800 ms
Initialization	Endpoint setup time	≤ 800-1200 ms
RedeemToken	Token redemption response	≤ 1200-1500 ms
ProvisionWallet	Wallet provisioning duration	≤ 1200-1500 ms
Authorization	Token validation turnaround	≤ 1200-1800 ms
CreatePass	Key creation and sync time	≤ 3 sec
All Actions	Failure rate threshold	≤ 1% in any rolling 15-minute window
All Actions	Log ingestion latency (Grafana visibility)	≤ 30 seconds (configurable) from event generation (depending upon KastleSDK's internet accessibility)

Dormakaba

Action	Performance Metric	Requirement
Authenticate	Token request and fallback resolution	≤ 1500-1800 ms (including fallback if needed)
SDK Initialization	SDK boot and config load	≤ 1200-1400 ms
BackendRegistrationInitiation	API registration method detection	≤ 1200-1500 ms

BackendRegistrationValidation	Token validation + backend acknowledgment	≤ 1200-1400 ms
CreateCredential	Requested (keyid) return	≤ 1000-1200 seconds
CreateCredential	Sync confirmation (Success)	≤ 15 minutes
CredentialStatus	Status change push from SDK	≤ 5 seconds
Unlock	Unlock callback timing	≤ 4-6 seconds (iOS) <= 5-8 seconds (android)
All Actions	Failure rate threshold	≤ 2% in rolling 15-minute window (≤ 2% for Unlock)
All Actions	Log sync and visualization lag	≤ 30 seconds (configurable) from each event generation (depending upon KastleSDK's internet accessibility)

Optimization Approaches:

[List approaches with descriptions]

Monitoring Strategy:

What	How
API/SDK Performance	Scalyr (request rates, durations, failure %)
Azure Function	Azure Monitor + App Insights (invocation counts, errors, cold starts), Log execution time, response codes, and retries
Cosmos DB	Azure Monitor (RU usage, throttling, latency, availability)

Availability & Recovery

Component	Strategy
SDK API Layer	Use **Azure App Service Premium** or **AKS with replica pods** behind a load balancer
Azure Functions (Logger Service)	Deployed in **multiple regions** (active-active or active-passive) using Azure Front Door or Traffic Manager
Cosmos DB	Enable **multi-region writes** and **geo-redundancy** with automatic failover
Grafana	Host on a **redundant infrastructure**, preferably via a managed Grafana service

Monitoring & Observability

Concept	Purpose	Tools

Monitoring	Real-time tracking of known issues (e.g., CPU, errors)	Grafana, Azure Monitor, Alerts
Observability	Ability to understand **why** things happen using metrics, logs, and traces	Logs in CosmosDB, distributed tracing, APM

Development Architecture

Dev Architecture Highlights

Layer	Tech Stack	Role
Mobile	Kotlin/Swift	User-facing logic
SDK	Kotlin/Swift + Native	Lock control & logging
API	.Net Core	Middleware & validation
Logger	Azure Function	Scalable ingestion
Storage	Azure Cosmos DB	Log persistence
Monitoring	Grafana	Visualization & alerts

Build & Test

Build Process:	Azure DevOps
Testing Strategy:	Automated Testing

Risks and Mitigations

Risk	Impact	Probability	Mitigation
Build pipeline failure	Delays in delivery, broken deployments	Medium	Enforce lint/tests in Azure DevOps; use gated/staged environments
Incomplete test coverage	Bugs in production, poor confidence in code	High	Define test strategy; enforce coverage thresholds; integrate tests in CI
Missing telemetry/logs in CosmosDB	Loss of audit trails, debugging difficulty	High	Add retry logic, schema validation, and monitoring on log writes
CosmosDB RU limits exceeded	API failures, slow performance	Medium	Enable autoscale, monitor RU consumption, batch writes

Grafana alert misconfiguration	Delayed incident response	Medium	Test alerts periodically, use thresholds + test notification channels
SDK/backend integration issues	Failed door unlocks or registrations	Medium	Add E2E tests; use mocks/stubs in test environments; add observability
Hardcoded or leaked secrets	Security breach, data exposure	High	Use Azure Key Vault and CI/CD secret scanning tools
Breaking changes in SDK/API	Downstream failures, client app crashes	Medium	Use semantic versioning, backward compatibility tests, release notes
Logs failed in push to final storage	Logging data missing	Medium	Device should save the logs until pushed successfully to final storage

---

Future Considerations

We may think changing the final storage, it may be prometheus.

Appendices

Technology Stack

Layer	Tech Stack	Role
Mobile	Kotlin/Swift/reactNative	User-facing logic
SDK	Kotlin/Swift + Native	Lock control & logging
API	.Net Core	Middleware & validation
Logger	Azure Function	Scalable ingestion
Storage	Azure Cosmos DB	Log persistence
Monitoring	Grafana	Visualization & alerts

Reference Documents

[List related documentation with document IDs and locations]

Glossary

Term	Definition
[Term]	[Definition]