NOWFLOATS™

Boost Technical Design Document

Document History

Date	Version	Author	Description
17/05/2022	0.1	Palwinder Singh	Initial Draft
19/05/2022	0.2	Vishal Sharma	Added more details around Viewpoints and format the document.

Approvers and Reviewers

Name	Role/ Organisation	Approver/Reviewer
Vishal Sharma	Head of Solution Architect	Approver

Introduction

Purpose:

This document defines boost360 architectural flow. Boost 360 is an online website builder which is designed for people who need a business website, but don't know how to code. This has a website make which comes with ready made website templates which are ideal for business owners who want a well-designed and fully functional website with minimum cost and effort.

Approach

NowFloats adopts an approach to architecture design based on viewpoints and perspectives.

Viewpoints are effectively layers of a system and its objectives. Viewpoints should be explainable in their own right. Viewpoints are:

- Functional
- Development
- Deployment

Perspectives cut across all the viewpoints and can also be considered as Non-Functional Requirements.

Perspectives are:

- Security
- Performance and Scalability
- Availability and Resilience
- Regulation
- Evolution
- Resource and Cost

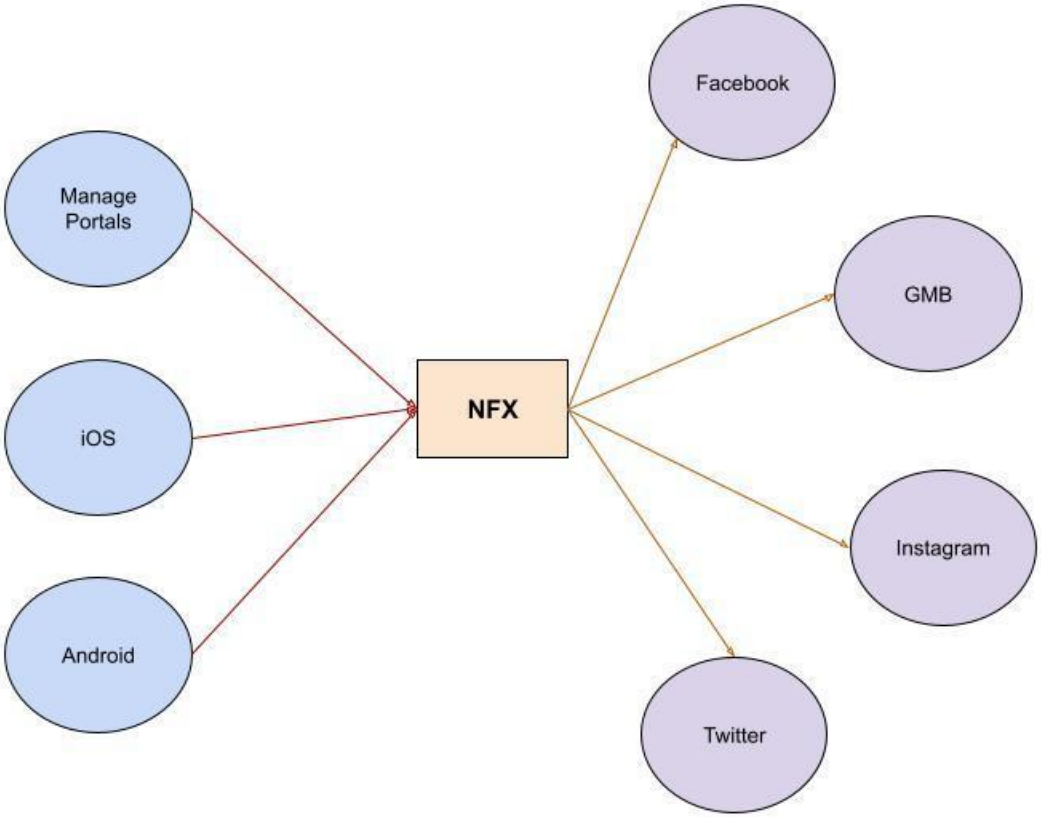
Architectural principles:

1. Deliver a great experience
2. Protect customer/vendor privacy
3. Enable vendors to deliver on the business case
4. Reuse where possible and don't reinvent the wheel
5. Promote ease of maintenance

Functional viewpoint

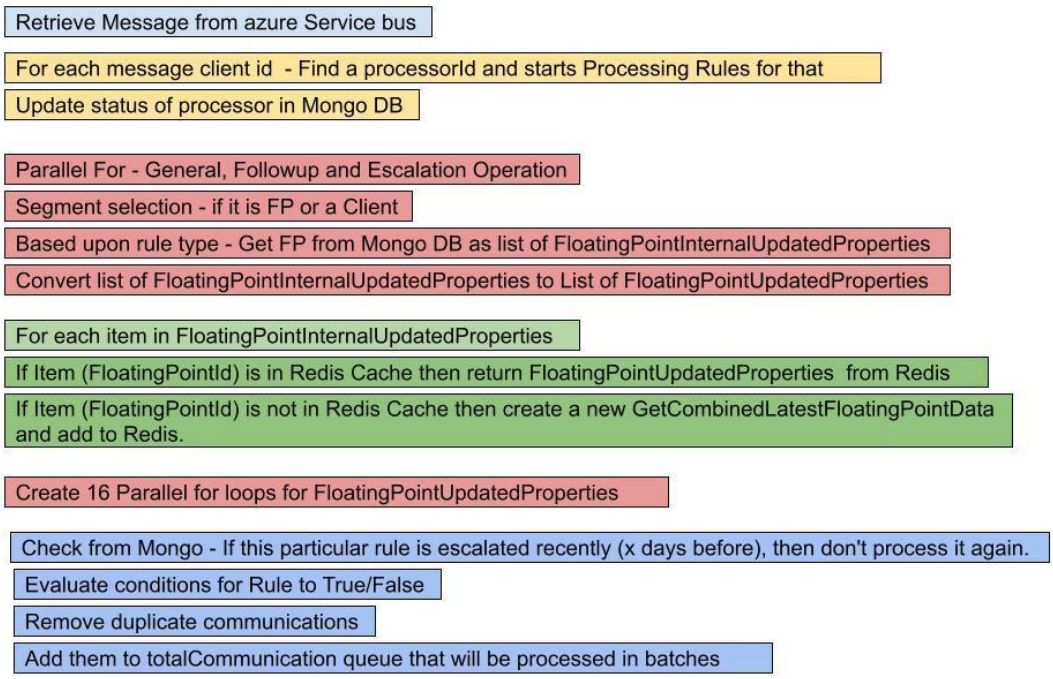
NFX:

NowFloats exchange is a system wherein NowFloats communicates with external entities like Facebook, Twitter, Google etc. This system is to streamline all the data flow both into and out of NowFloats.



Ria:

This service sends out notifications to users based on some predefined business rules. Here is high level flow diagram for Ria rule processor.



Boost Web:

Boost web is portal which acts as a CMS for merchant website. There are lot of features which a merchant can use as a service or as a product for his category. This is also available on android and IOS.

Merchant website:

This is actual website for merchant which is front end of boost CMS. There are predefined categories and themes which user can select during registration for his website.

Communication services:

Communication services are common services which are making / sending notifications and messages across boost platform. They send out sms, whatsapp notifications, push notifications and emails to customers.

Development ViewPoint

The technical stack is as follows:

Component	Technologies
Database	MongoDB, MySQL,SQL server, Firestore DB
Microsoft .NET Framework	ASP.net core 2.0
Microsoft Development Language	C#, Nodejs 6.0, Knockout js, Angular
Other languages	Python, Golang
Operating System	Windows, Android, IOS
Cloud providers	AWS, Azure, GCP
Reporting	Tableau
BI data warehouse	BigQuery

Application Design Principles

Separation of concerns

The application has been divided into distinct features/components/services with as little overlap in functionality as possible. Each service has deployed as independent component, any other application can reuse business components without going through the front layer and the presentation layer can easily be enhanced/ replaced without impacting the business layer and vice versa.

Single responsibility principle

Each component would be responsible for only a specific feature or functionality, or aggregation of cohesive functionality. This principle in our application design it helped to produce more loosely coupled and modular systems. After application of this principle to application architecture and taken to its logical endpoint, we get microservices. A given microservice should have a single responsibility. If we need to extend the behavior of a

system, it's usually better to do it by adding additional microservices, rather than by adding responsibility to an existing one.

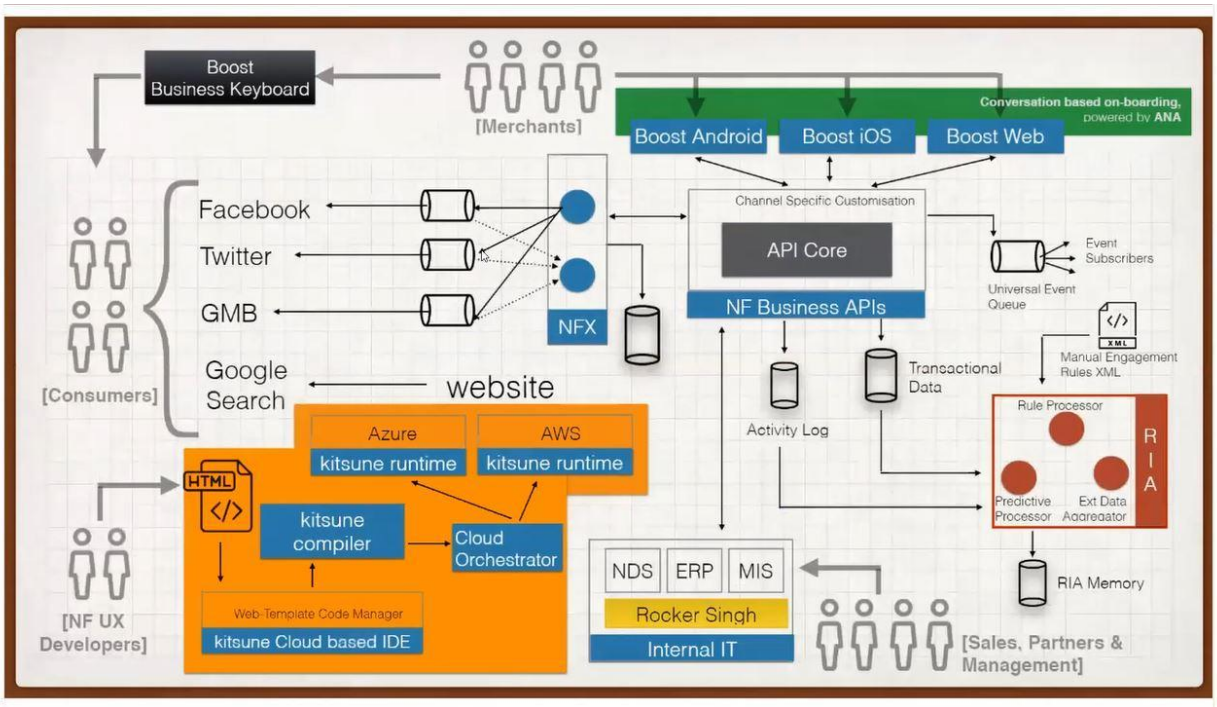
Principle of Least Knowledge

In order to ensure extensibility a component or object would not know about internal details of other components or object, thus reducing the coupling between the components.

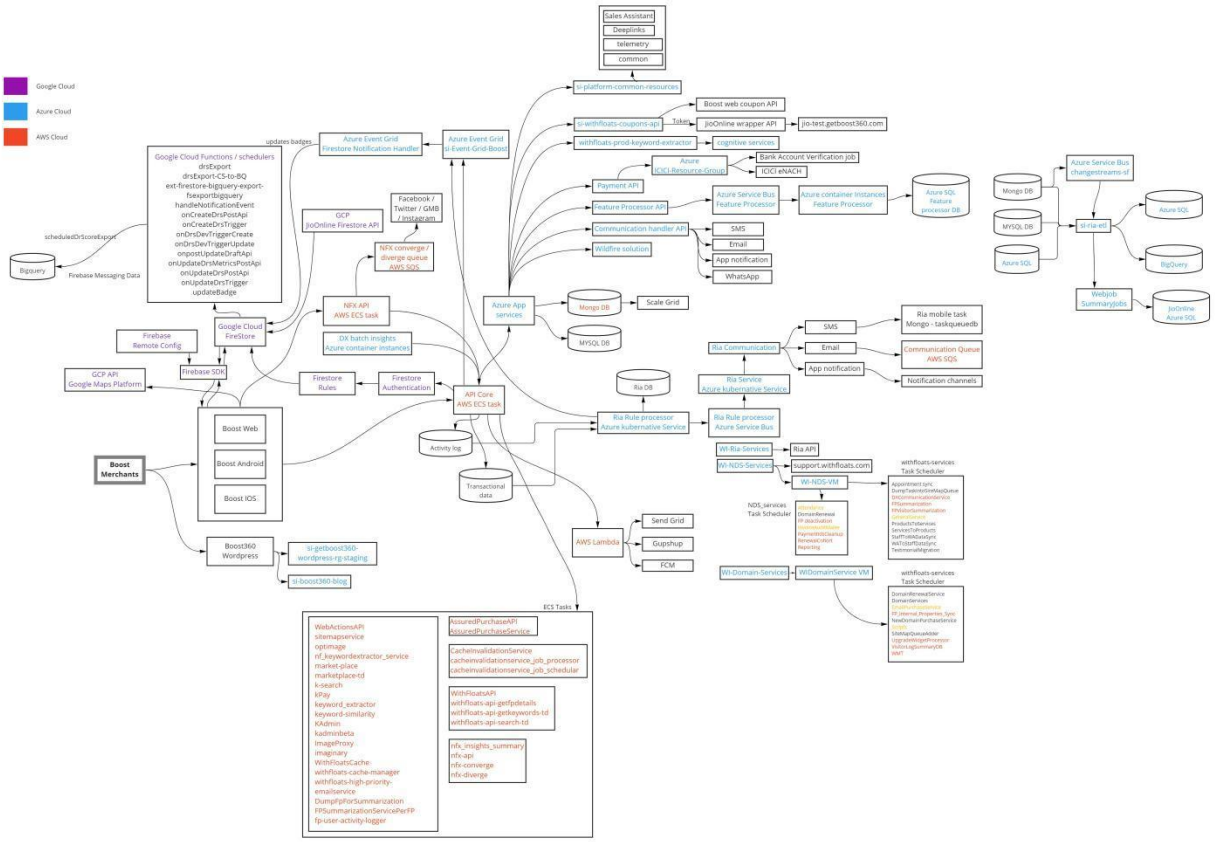
Cross-cutting concerns

Mixing the code that implements these functions with the business logic can lead to a design that is difficult to extend and maintain. Use of techniques like aspect oriented programming would be considered to achieve this.

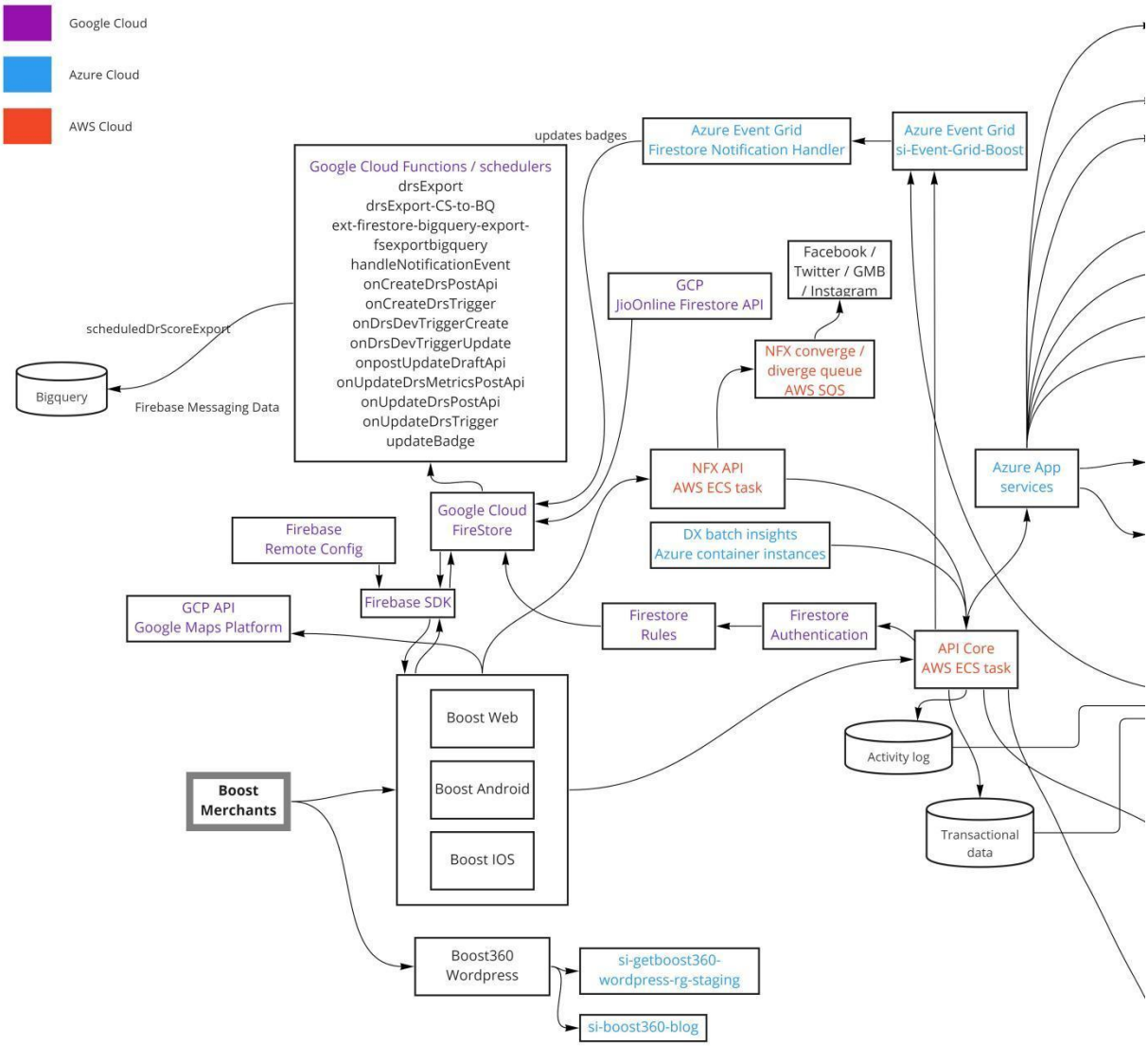
High Level Component Design Architecture

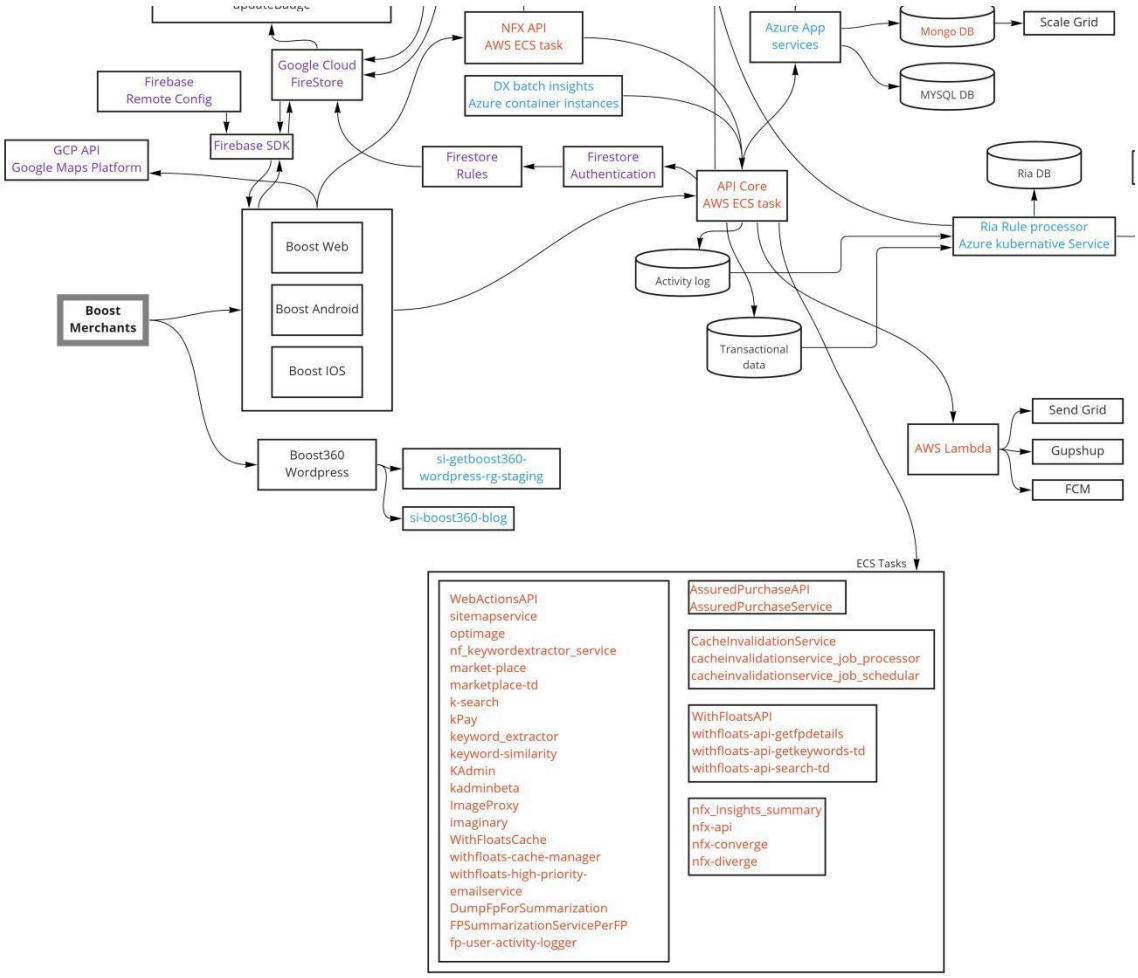


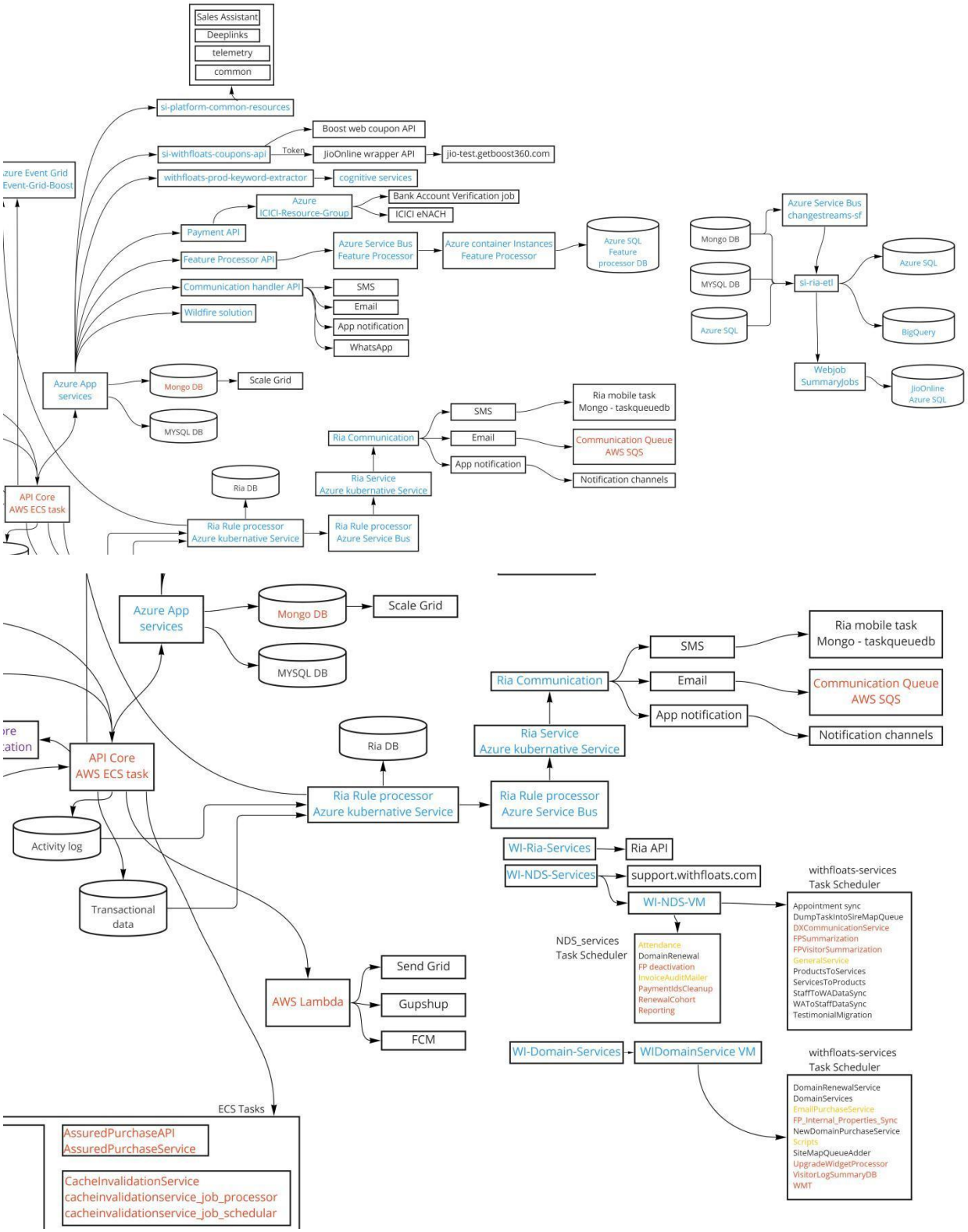
Low Level Component Design Architecture



Here are zoomed versions of above diagram:




NFX API
AWS ECS task
Azure App services
Mongo DB
Scale Grid
Google Cloud FireStore
Firebase Remote Config
DX batch insights
Azure container instances
MYSQL DB
Firebase SDK
GCP API
Google Maps Platform
Firestore Rules
Firestore Authentication
API Core
AWS ECS task
Ria DB
Ria Rule processor
Azure kubernative Service
Boost Web
Boost Android
Boost IOS
Boost Merchants
Activity log
Transactional data
Send Grid
AWS Lambda
Gupshup
FCM
Boost360 Wordpress
si-getboost360-wordpress-rg-staging
si-boost360-blog
ECS Tasks
WebActionsAPI
sitemapservice
optimage
nf_keywordextractor_service
market-place
marketplace-td
k-search
kPay
keyword_extractor
keyword-similarity
KAdmin
kadminbeta
ImageProxy
imaginary
WithFloatsCache
withfloats-cache-manager
withfloats-high-priority-emailservice
DumpFpForSummarization
FPSummarizationServicePerFP
fp-user-activity-logger
AssuredPurchaseAPI
AssuredPurchaseService
CacheInvalidationService
cacheinvalidationservice_job_processor
cacheinvalidationservice_job_schedular
WithFloatsAPI
withfloats-api-getfpdetails
withfloats-api-getkeywords-td
withfloats-api-search-td
nfx_insights_summary
nfx-api
nfx-converge
nfx-diverge


Sales Assistant
Deeplinks
telemetry
common
si-platform-common-resources
Boost web coupon API
si-withfloats-coupons-api
Token
JioOnline wrapper API
jio-test.getboost360.com
withfloats-prod-keyword-extractor
cognitive services
Azure ICICI-Resource-Group
Bank Account Verification job
ICICI eNACH
Payment API
Feature Processor API
Azure Service Bus Feature Processor
Azure container Instances Feature Processor
Azure SQL Feature processor DB
Communication handler API
SMS
Email
App notification
WhatsApp
Wildfire solution
Azure App services
Mongo DB
Scale Grid
MYSQL DB
Ria Communication
SMS
Email
App notification
Ria mobile task Mongo - taskqueuedb
Communication Queue AWS SQS
Notification channels
Ria Service Azure kubernative Service
Ria DB
API Core AWS ECS task
Ria Rule processor Azure kubernative Service
Ria Rule processor Azure Service Bus
Mongo DB
MYSQL DB
Azure SQL
Azure Service Bus changestreams-sf
si-ria-etl
Azure SQL
BigQuery
Webjob SummaryJobs
JioOnline Azure SQL
Azure App services
Mongo DB
Scale Grid
MYSQL DB
Ria Communication
SMS
Email
App notification
Ria mobile task Mongo - taskqueuedb
Communication Queue AWS SQS
Notification channels
Ria DB
Ria Service Azure kubernative Service
API Core AWS ECS task
Ria Rule processor Azure kubernative Service
Ria Rule processor Azure Service Bus
Activity log
Transactional data
WI-Ria-Services
Ria API
WI-NDS-Services
support.withfloats.com
WI-NDS-VM
withfloats-services Task Scheduler
Appointment sync
DumpTaskIntoSireMapQueue
DXCommunicationService
FPSummarization
FPVisitorSummarization
GeneralService
ProductsToServices
ServicesToProducts
StaffToWADataSync
WAToStaffDataSync
TestimonialMigration
NDS_services Task Scheduler
Attendance
DomainRenewal
FP deactivation
InvoiceAuditMailer
PaymentIdsCleanup
RenewalCohort
Reporting
AWS Lambda
Send Grid
Gupshup
FCM
WI-Domain-Services
WIDomainService VM
withfloats-services Task Scheduler
DomainRenewalService
DomainServices
EmailPurchaseService
FP_Internal_Properties_Sync
NewDomainPurchaseService
Scripts
SiteMapQueueAdder
UpgradeWidgetProcessor
VisitorLogSummaryDB
WMT
ECS Tasks
AssuredPurchaseAPI
AssuredPurchaseService
CacheInvalidationService
cacheinvalidationservice_job_processor
cacheinvalidationservice_job_schedular

Components details:

Boost web/Android/IOS:
This is a boost client on web/Android/IOS app which generates merchant requests for handling merchant website. Boostweb is angular website with node js and have services in C#.net core.

Withfloat APIs:
These are core APIs which are called by Boost ecosystem and shared across all the platforms. This is a layer where an actual business logic is written.

Ria Rule processor :
This is business logic scheduler in which business rules have been written and they acre scheduled to be run daily, weekly and monthly basis. These rules execute for each FP and website and they send out notifications in case of any rule fails on any FP.

Ria Communication:
This is communication layer which basically sends out notifications like email, sms, push and whatsapp to different users. Ria rule processor or many Boost services uses these notification APIs.

NFX/DX services:
These are services wherein NowFloats communicates with external entities like Facebook, Twitter, Google etc. This system is to streamline all the data flow both into and out of NowFloats. Basically NF publishes then contents onto social media channels using these APIs.

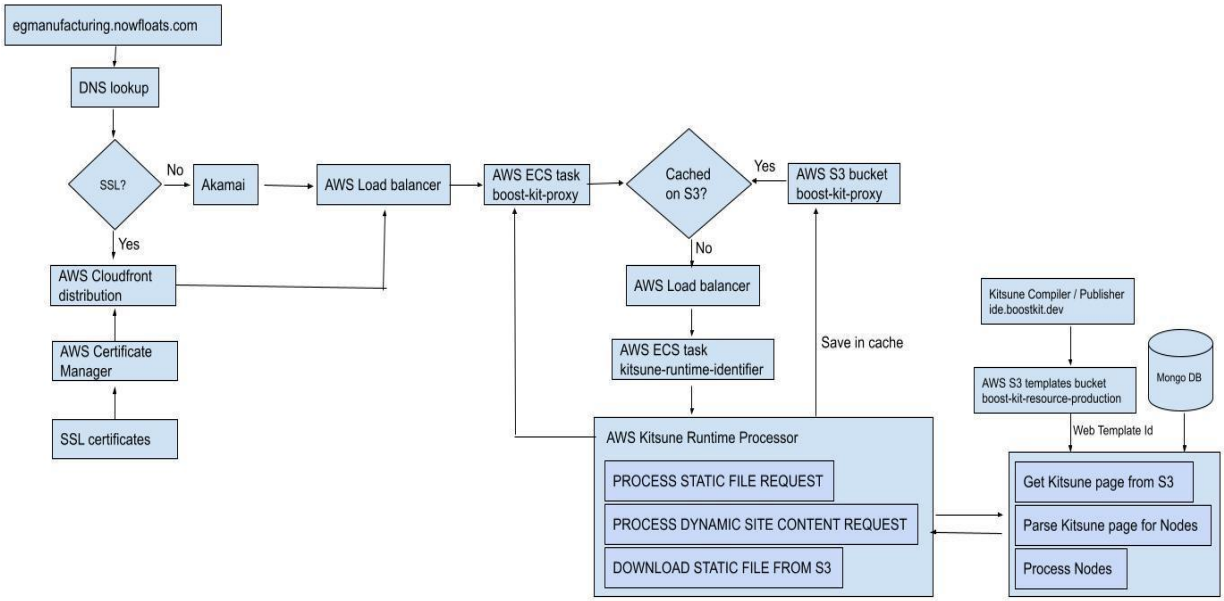
Payment API:
These APIs are for payment processing. They generates payment links like from Razor pay and then web hooks are written in NF APIs to cater order processing further after payment API returns success message.

Feature processor API :
These APIs manages features/ widgets which user has purchased for this website. These APIs checks validity and expiration of various packages for a user and it activates or deactivates packages based on user subscriptions.

Communication handler API:
Communication services are common services which are making / sending notifications and messages across boost platform. They send out sms, whatsapp notifications, push notifications and emails to customers.

Window services:
There are couple of windows task schedulers which are doing data transformations from various databases like MySQL, SQL server and Mongo DB.

Customer Website Flow Diagram



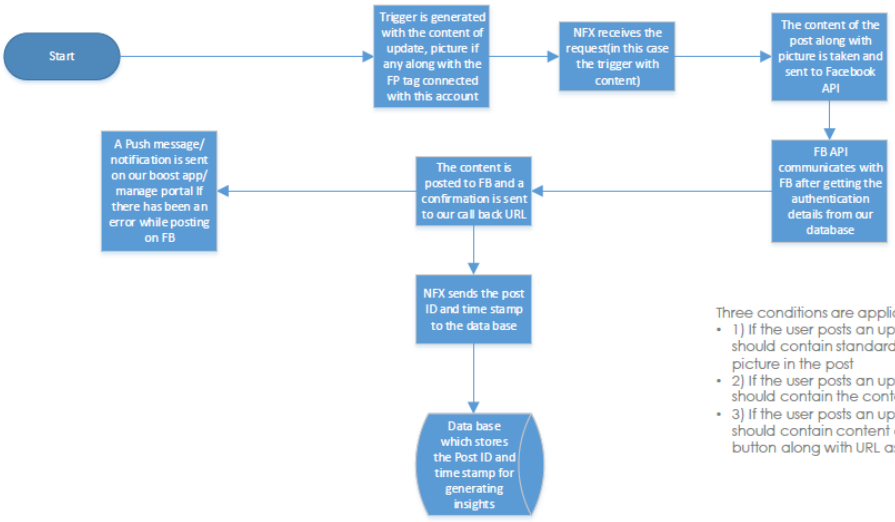
NFX Flow Diagram

NowFloats exchange is a system wherein NowFloats communicates with external entities like Facebook, Twitter, Google etc. Once this system is implemented we aim to streamline all the data flow both into and out of NowFloats.

Merchant Authentication – Our merchant allows us to post on his behalf

Continuous updates

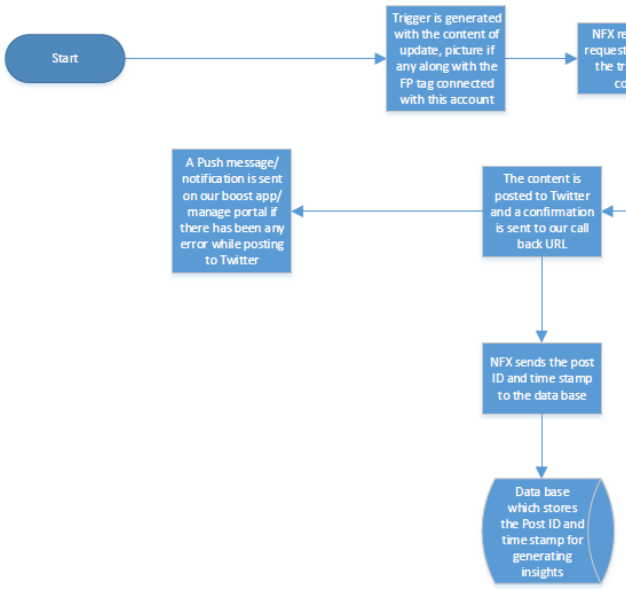
Facebook Flow



Three conditions are applicable here

- 1) If the user posts an update with picture then the facebook post should contain standard content (with URL as a part of text) and a picture in the post
- 2) If the user posts an update only with text then the Facebook post should contain the content along with URL passed as a link
- 3) If the user posts an update as a product then the facebook post should contain content and call to action button with the SHOP NOW button along with URL as a field

Twitter Flow

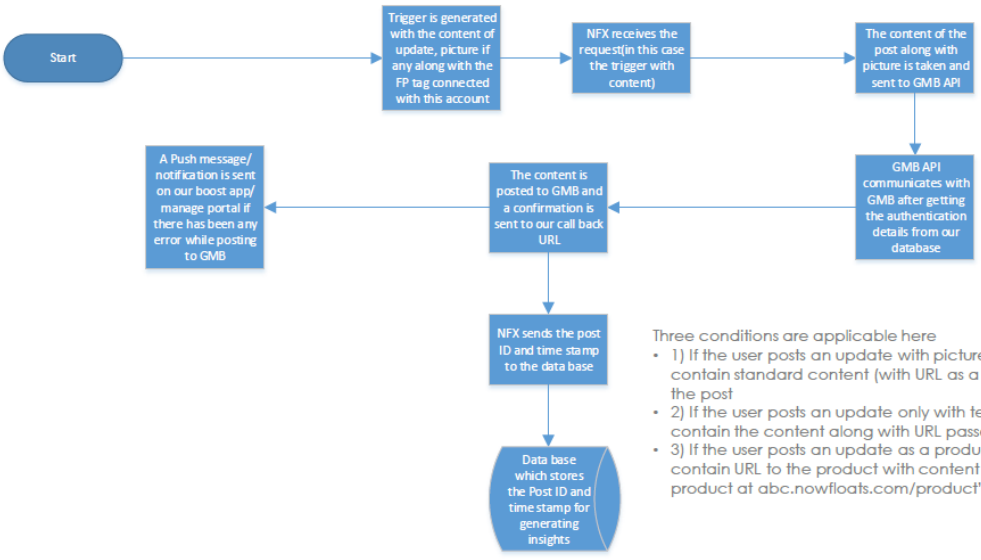


Three conditions are applicable here

- 1) If the user posts an update with picture then the twitter post should contain standard content (with URL as a part of text) and a picture in the post
- 2) If the user posts an update only with text then the Twitter post should contain the content along with URL passed as a link
- 3) If the user posts an update as a product then the Twitter post should contain URL to the product with content as "Check out our new product at abc.nowfloats.com/product"

*For the text – If text characters exceed 140 then post "Check out our new post at abc.nowfloats.com"

GMB Flow



Three conditions are applicable here

- 1) If the user posts an update with picture then the GMB post should contain standard content (with URL as a part of text) and a picture in the post
- 2) If the user posts an update only with text then the GMB post should contain the content along with URL passed as a link
- 3) If the user posts an update as a product then the GMB post should contain URL to the product with content as "Check out our new product at abc.nowfloats.com/product"

Common Components

Exception Handling, Logging, Instrumentation and Diagnostics:

Boost have following exception handling and logging mechanisms:

- AppInsight for logging Azure services
- Cloudwatch for logging AWS services
- Server logs for azure and AWS scheduled services on VMs
- Event logs for azure and AWS scheduled services on VMs
- Sentry logs for boost web, Android and IOS applications
- Some services like communication or legacy services, are also logging to MongoDB, MySQL db to respective log tables.

Development Standards

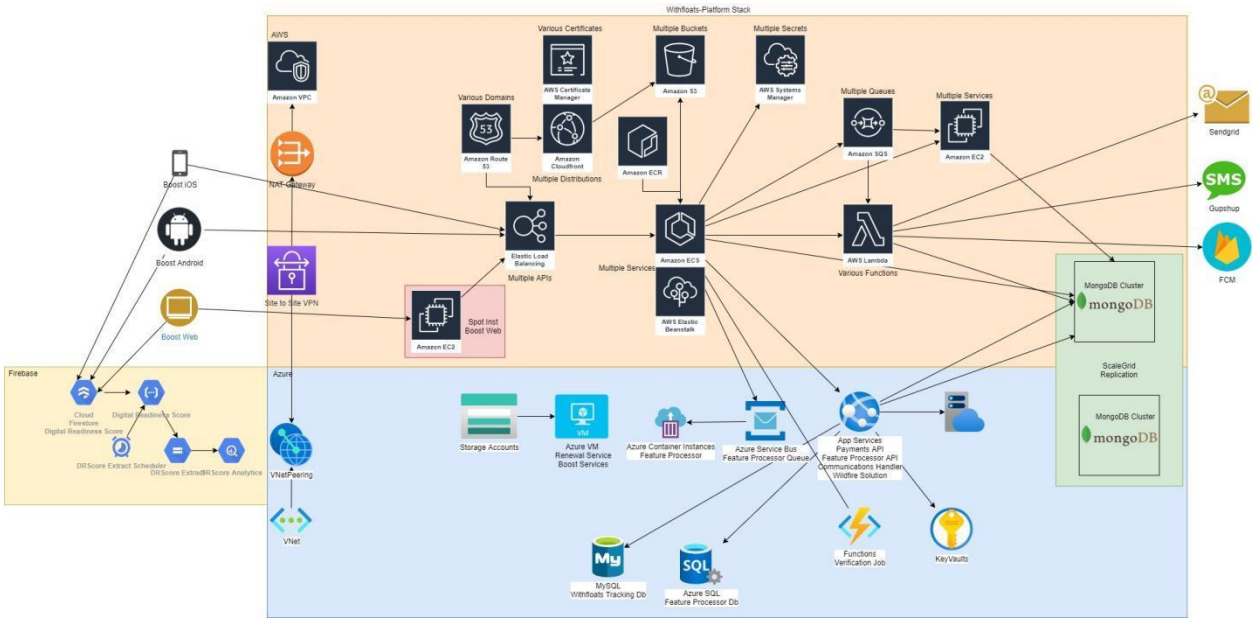
The development will adhere to standards and best practices recommended by Microsoft for development in general.

Finally, NowFloats will define a set of best practices, developer checklists and automated tools to assess code quality.

Examples include the use of Codacy.

Deployment ViewPoint

Infrastructure Diagram



Locations

Azure/AWS Location details.

Location	Geographical Locatoin
Azure	Azure Central India (Pune), South India (Chennai), and West India (Mumbai)
AWS	Asia Pacific (Mumbai) Region, Asia Pacific (Singapore)

Environment specifications

Environment List

The following environments may well be required. The exact number of test environments could vary from the model below, but this at least sets out an initial view.

Environment Name	Dev	Staging	Prod
Build	Y	Y	
Functional Test	Y	Y	
Deployment Test	Y	Y	
System Test		Y	Y
Pre-Production			
Production			

Risks

S.No	Risk	Severity
1	Boost only has a single environment, production. Development and production should be separate.	Severity 1
2	Test Data, development data should be separate out from production environment data	Severity 2
3	Deployment CICD should be employed for various environments	Severity 2
4	Boost is running on older kitsune. There are a lot of new languages out in the market which provide cutting edge features if we look boost with its similar competitors in the market. These languages provide out of box features for performance and scalability.	Severity 2