

Smartface

Smartface Middleware Platform

FEATURES & BENEFITS WHITEPAPER



SMARTFACE DIGITAL TRANSFORMATION PLATFORM

- NEXT GENERATION MIDDLEWARE & API MANAGEMENT
- MULTI-CHANNEL DEVELOPMENT PLATFORM
- MOBILE CI/CD PLATFORM
- ENTERPRISE MOBILE APP STORE
- MOCK API SUITE

THE PILLAR OF DIGITAL TRANSFORMATION

SMARTFACE MIDDLEWARE

Digital transformation in the enterprises with a well-established traditional and legacy software stack may face unpredictable challenges in making the new generation apps play well with the traditional systems.

With its next generation and futureproof architecture, Smartface Middleware Platform facilitates digital transformation by alleviating the complexities of connecting traditional architectures to the modern multi-channel frontends.

1

No More Cross Cutting Concerns

All the cross-cutting concerns (CCC) such as authentication, authorization, logging, exception logging, log analytics, configuration and secret management, caching, APIfication, API security, API message validation, API message transformation, API productizing, log analytics, metrics analytics, business process management, business rules management, etc. can be fully offloaded to the middleware components.

No need for custom development or third-party apps for each CCC along with centralization.

2

No More High Cost Servers

With the CCC offloading and 12-Factor App microservice architecture, any app can be transformed to a containerized microservice application.

The transformed cloud native app can be run on any Kubernetes or OpenShift environment in a lightweight, highly available and clustered manner **with higher availability, better performance and lower cost than the traditional app servers.**

3

No More Legacy Vendor Lock-in

Any custom components, customizations or extensions developed in the legacy backend systems can be offloaded to the middleware.

Once the rule-based composition logic is separated, the remaining codebase can be transformed to microservices to be run in a cloud native architecture on OpenShift or Kubernetes. Similarly, current legacy backend functions can be transformed into APIs for external consumption. After these transformations, **the legacy backend systems can be fully isolated from the custom developments** for better maintainability and if needed, replaceability.

4

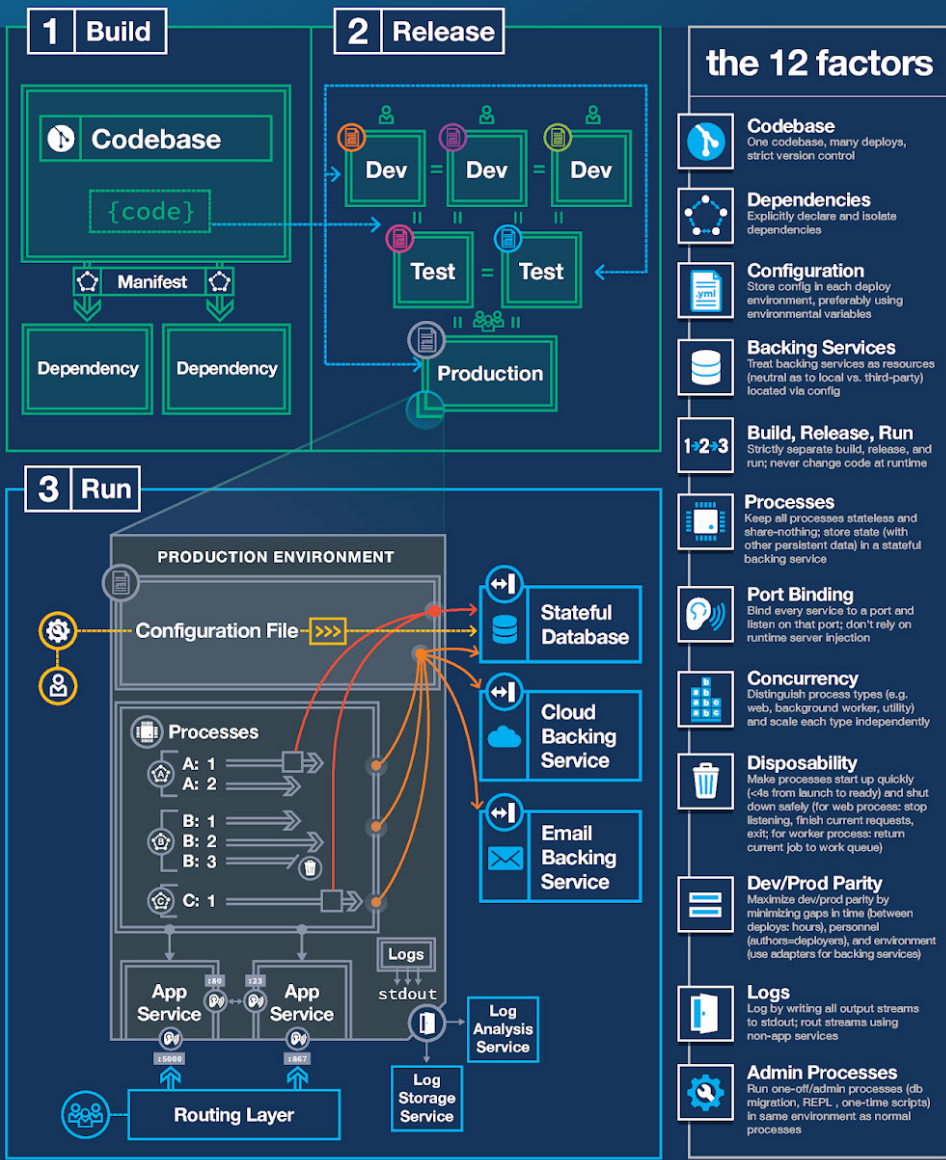
Reduce Developer Dependency

Instead of a single, colossal custom code block maintained by a single “superstar” developer, the functionality is implemented in the middleware layer with the best practices and the right architecture.

You can divide and conquer the dependency on individual developers with the best practices.

The 12-Factor App

Modern web applications run in heterogeneous environments, scale elastically, update frequently, and depend on independently deployed backing services. Modern application architectures and development practices must be designed accordingly. The PaaS-masters at Heroku summarized lessons learned from building hundreds of cloud-native applications into the twelve factors visualized below.



5 Empowering Citizen "Rulers"

All the business rules are strictly isolated from the code and developed within the rule engine, reducing maintenance efforts, enabling better auditing and reporting and **empowering analyst-level, citizen resources for rule management** instead of developers.

Smartface Middleware Platform is developed with the 12-Factor App Approach for Cloud Native Apps

6 Benefit from the API Economy

The API Management component in the middleware allows the **commercialization and the productization of the APIs** for an internal or an external product ecosystem.

API economy requires providing APIs with high availability and performance within SLAs and the Smartface Middleware Platform is developed with a best practice architecture for the highest SLAs.

7 Real-time Analytics

With the API Gateway logging all API traffic and payload data in the log server, it is possible to get **real-time API data analytics reports from the middleware** instantly in a self-service manner, reducing the log report requests from the enterprise data warehouse and the reporting team.

Moreover, with the CCC minimizing components offloaded to the middleware, **the functionality logs are also unified**, enabling individual component-based as well as cross-referenced availability, performance and troubleshooting analyses.

8 Visual Metric Analytics

With the readily available metrics in the CCC components and the microservice metrics mandated by the 12-Factor App Approach, all middleware metrics are collected in the time series database component and **presented in smart and functional dashboards**. Monitor your metrics for improvement.

9

No More Database Vendor Lock-in

With the 12-Factor App microservice architecture, Smartface Middleware enables and facilitates **switching from expensive and locked-in proprietary database solutions to open source database solutions**, reducing costs and eliminating dependencies.

10

No More Single Point of Failure

Instead of being forced to run a single instance due to hardware, license, cost or resource limitations of the traditional legacy systems, all apps transformed to the microservices with the middleware and the 12-Factor App approach can be run in multiple instances on the OpenShift or Kubernetes containers.

Depending on the current load, **the number of running instances are automatically upscaled or downscaled dynamically**, optimizing the resource usage to its fullest.

11

High Compatibility

The cloud native middleware solution is **compatible with OpenShift and Kubernetes**. In whichever technology the enterprise has invested, the investment can be preserved.

12

Rapid Delivery and Productization

With the middleware solution providing an extensive number of functionalities in a unified manner, any newly developed app will not require the rediscovery and reimplementations of these common functionalities.

This facilitates the **rapid delivery of apps along with the rapid productization and market acceptance**, as the middleware will have proven itself in previous apps.

13

Enable Frontend Channels Rapidly

Most of the traditional legacy systems still rely on SOAP or ancient connectivity technologies; however especially for the mobile apps and the next generation frontends, it is crucial to use the REST or gRPC connectivity standards for higher performance and higher reliability.

With the API management components in the middleware, **SOAP to REST mediation can be handled in the middleware layer without any changes in the legacy backend systems** to provide optimized REST APIs to the frontend channels.

WHY SMARTFACE

Smartface: The Cloud Pioneer of Enterprise Mobility

Smartface provides a fully cloud-based, cloud native, continuously integrated platform with multi-channel app development & lifecycle management and middleware & API management. With Smartface, it is possible to develop native iOS/Android mobile and single-page web applications and mobile backends just with JavaScript/TypeScript knowledge and centrally manage all enterprise mobility and frontend management processes. The platform can run on compatible Kubernetes or OpenShift environments for the maximum flexibility.