

AGL Navi-EG

Yoshito Momiyama
2017.2.23

概要

AGLのNavi-API策定にあたり、ベースにするGENIVI APIが前提しているアーキテクチャを分析し、AGLのNavi-APIの方向性を決める材料にする

Navi-EG Goal

AGL Apps.

Navi Application

POI Application

AGL Defined API

GENIVI-API

Extended by Navi-EG

AGL Navigation API

Vendor Extended API

AGL Compliant Navi engine

OSS Navi
(gps-navi)

Selectable

Product Navi

AGL Device API

Devices/Middleware

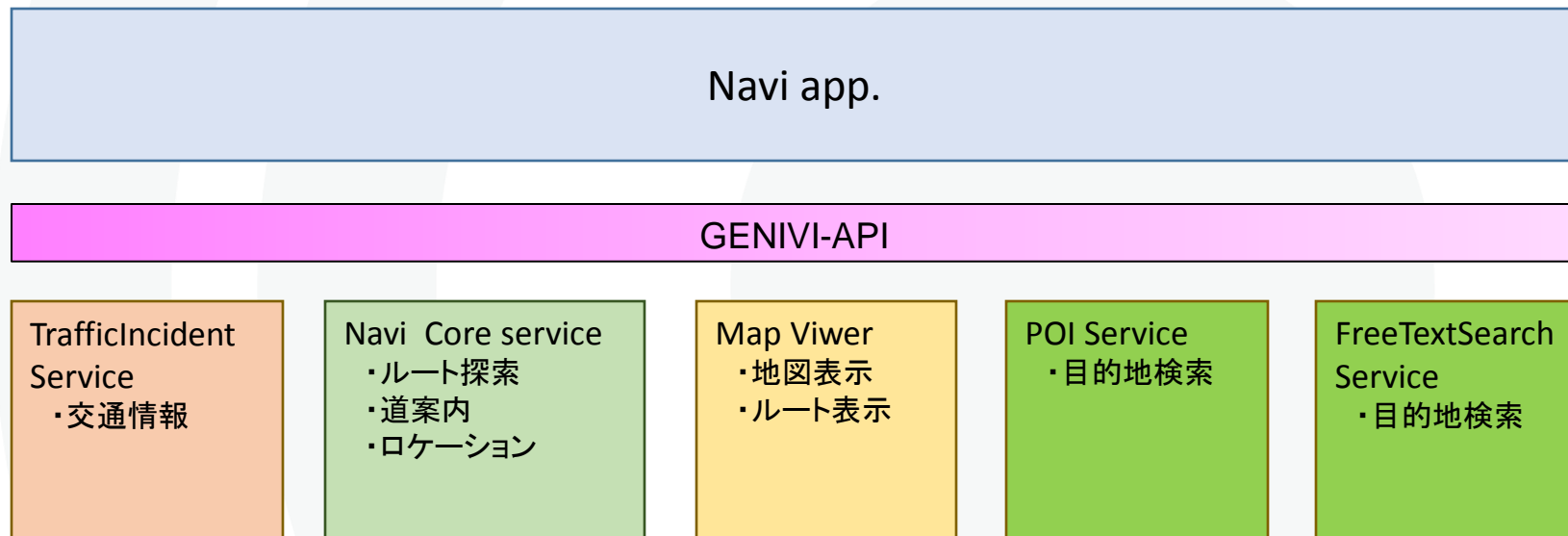
(1) AGL provides common Navi API.

(2) AGL scalability should be allowed as long as compatibility allows

(3) AGL provides P/F which allows to select OSS or Product Navi

(4) AGL provides P/F which allows to select OSS or Product Navi

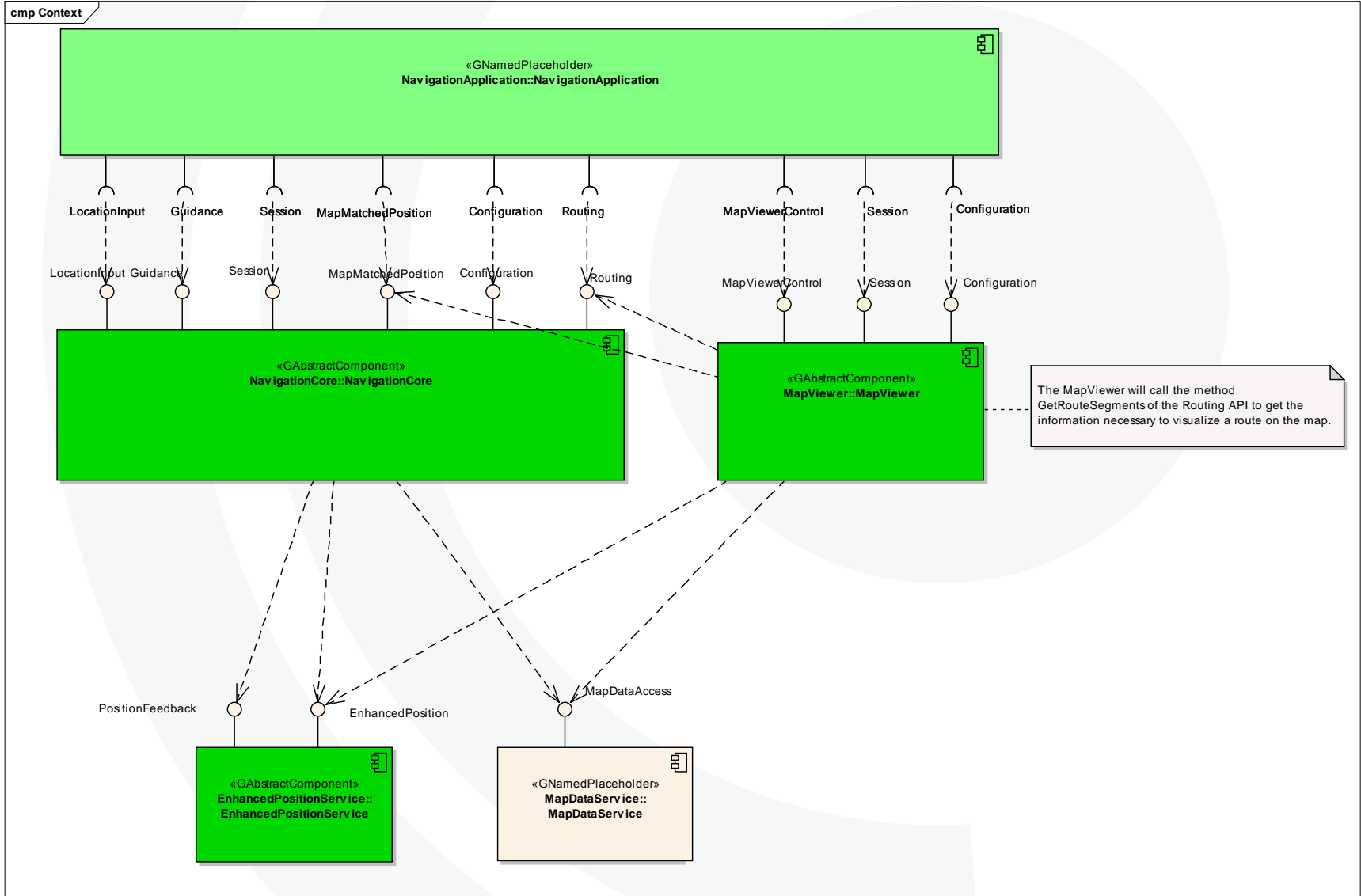
GENIVIのアーキテクチャ(全体)



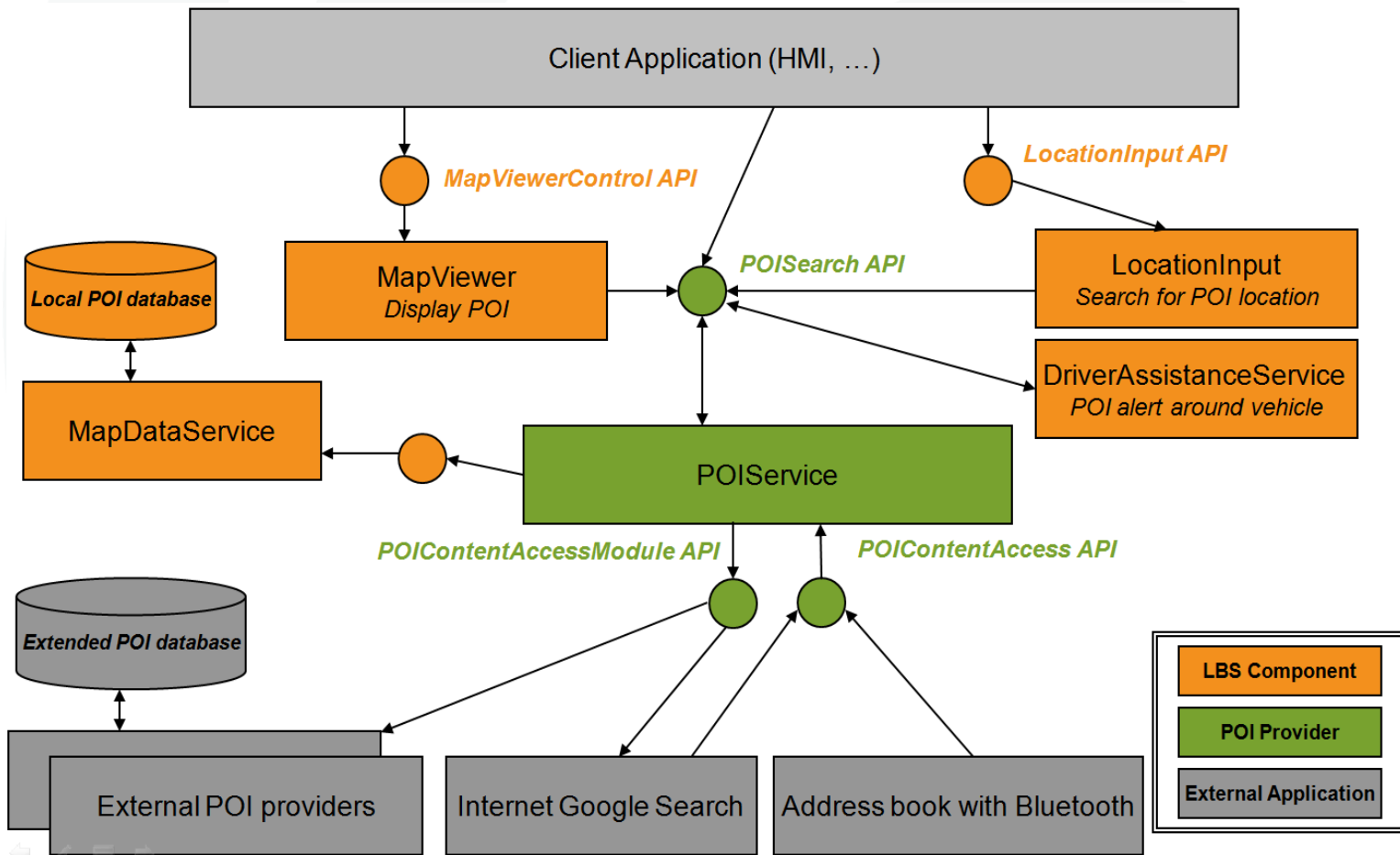
GENIVIのアーキテクチャは、強い連携が必要なナビの基本機能(ルート探索、道案内、ロケーション)をまとめてcore serviceとして扱っており、地図表示・ルート表示をMapViwerが担う。

交通情報や目的地検索に関しては、コアサービスには含まれず、サービス単位で分割する構成になっている。

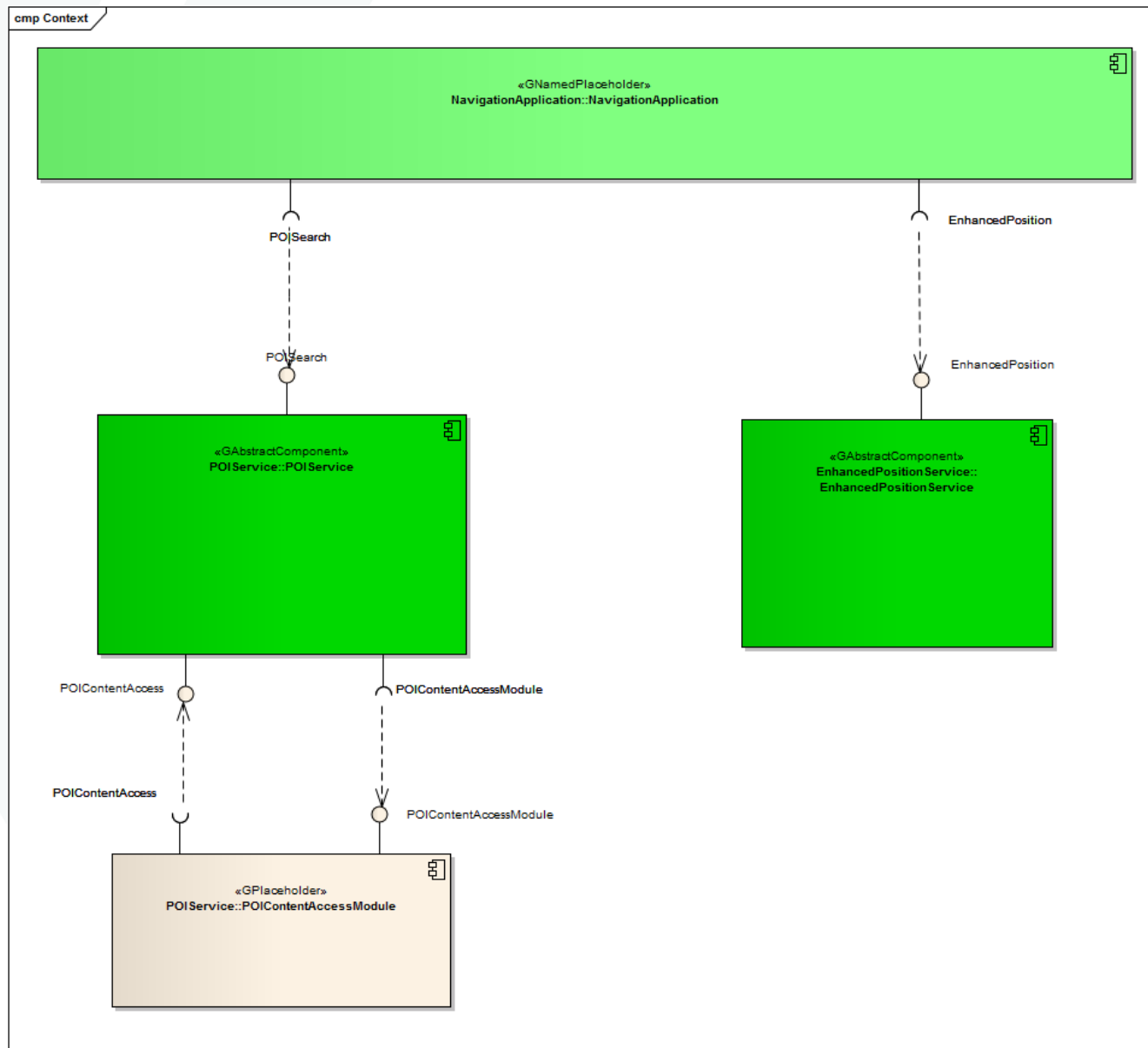
GENIVIのアーキテクチャ(地図・ナビコア)



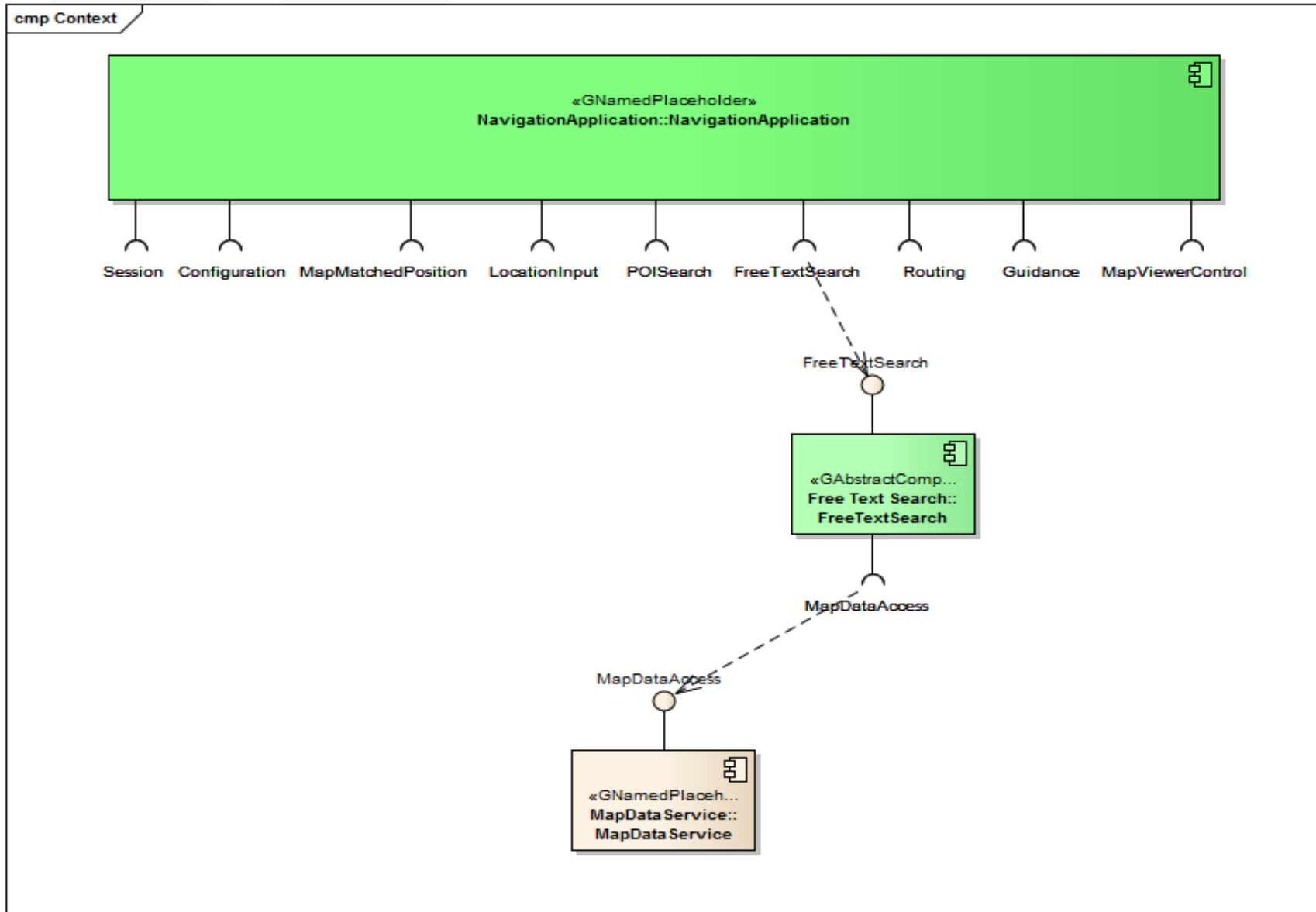
GENIVIのアーキテクチャ(POI-1)



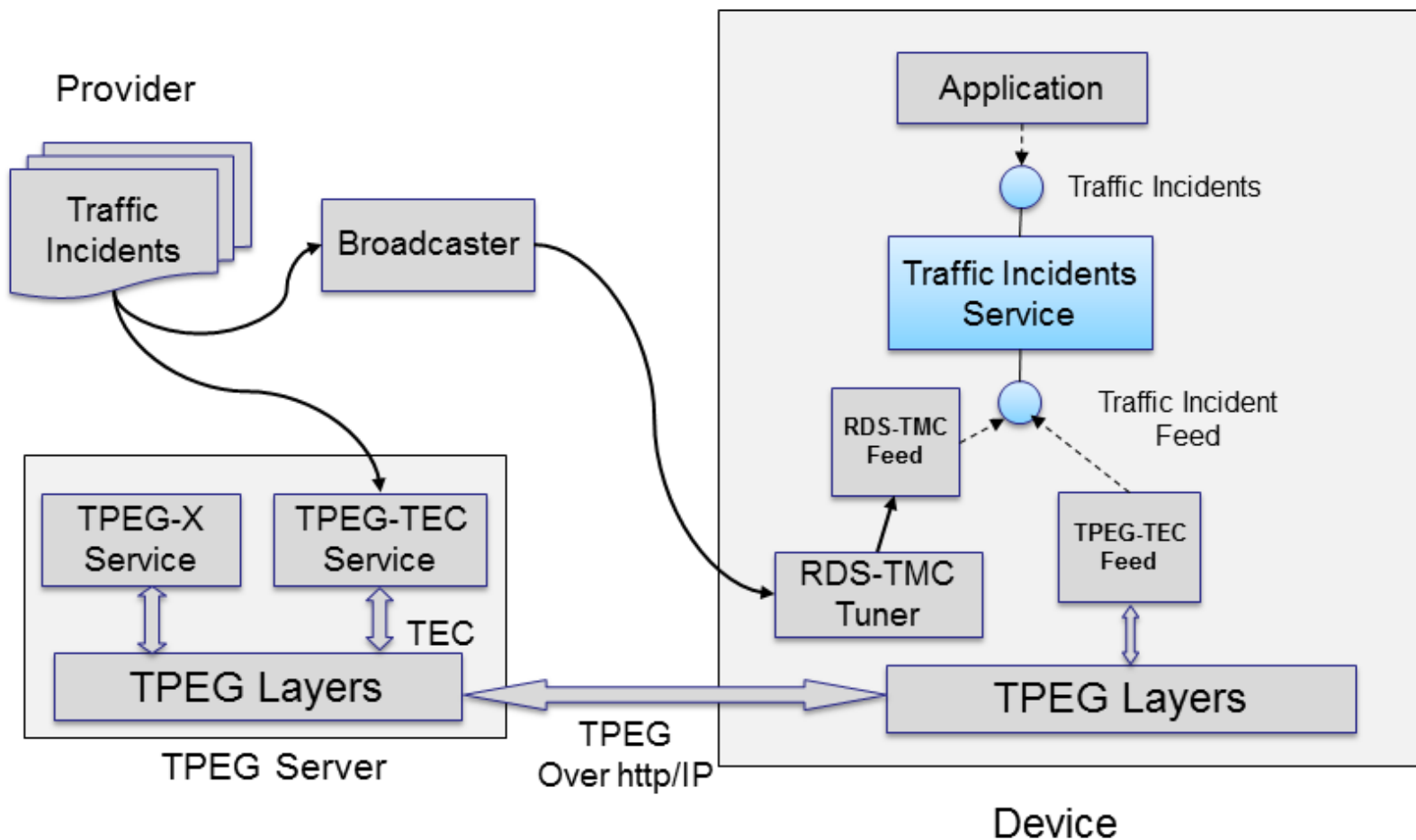
GENIVIのアーキテクチャ(POI-2)



GENIVIのアーキテクチャ(FreeTextSearch)

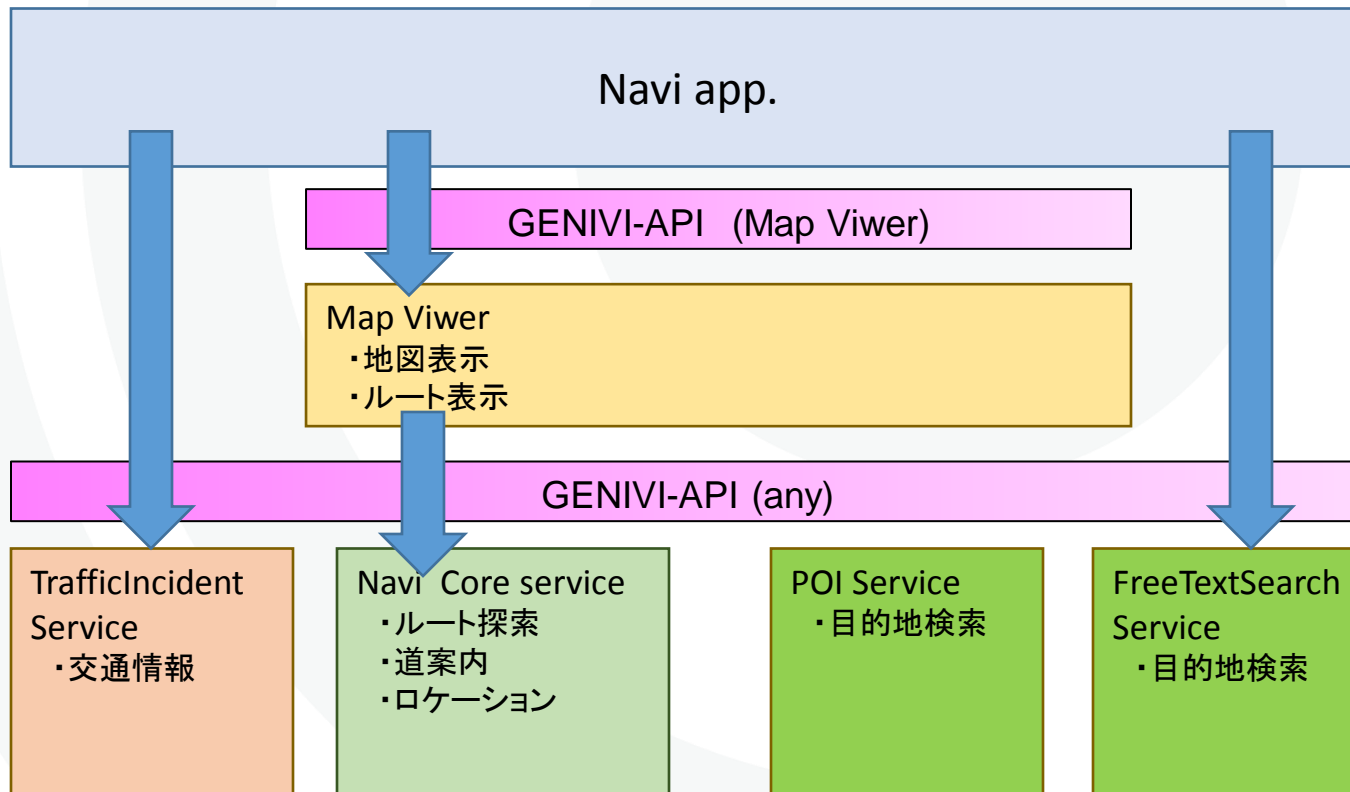


GENIVIのアーキテクチャ(FreeTextSearch)



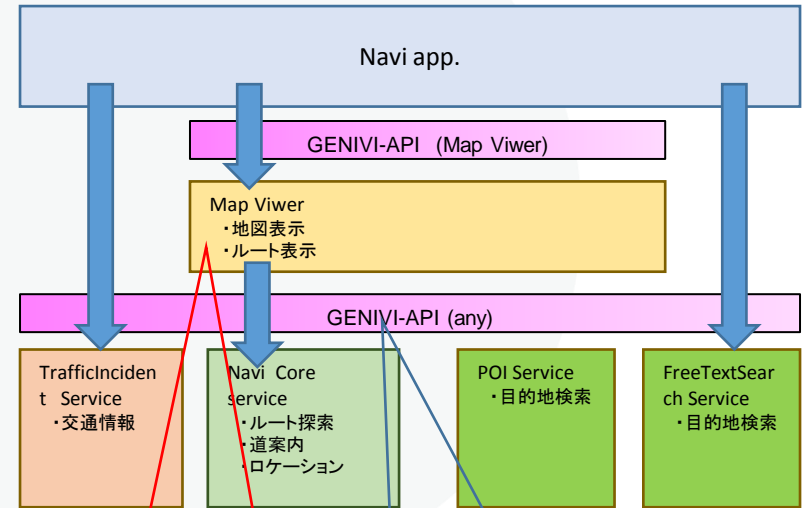
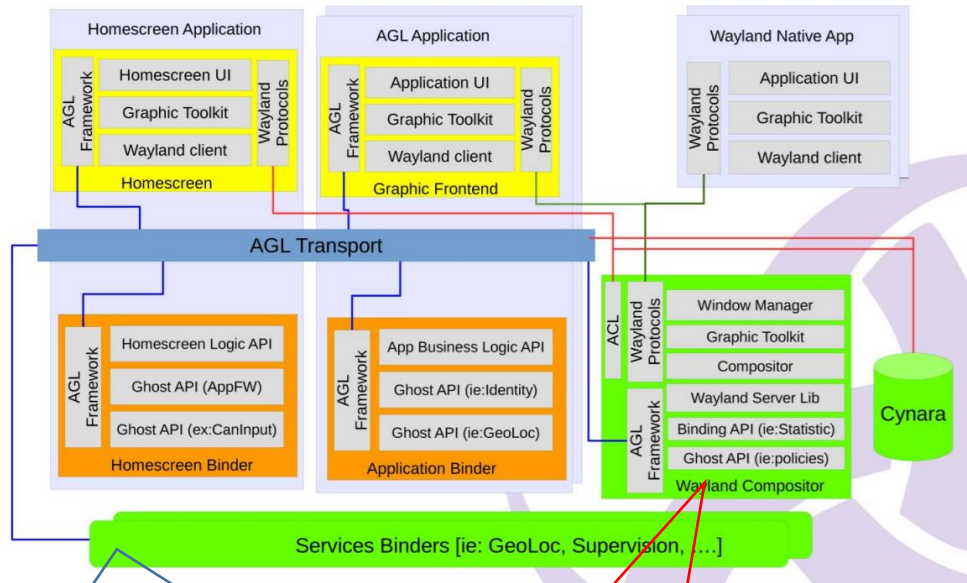
GENIVIのアーキテクチャ(全体-2)

各サービスのアーキテクチャを元にするると、実際のアーキテクチャはMap Viwerが一段上に乗る形になっていると考えたほうが良い



AGLにおけるGENIVIアーキテクチャの課題

AGL-DD Proposed Logic



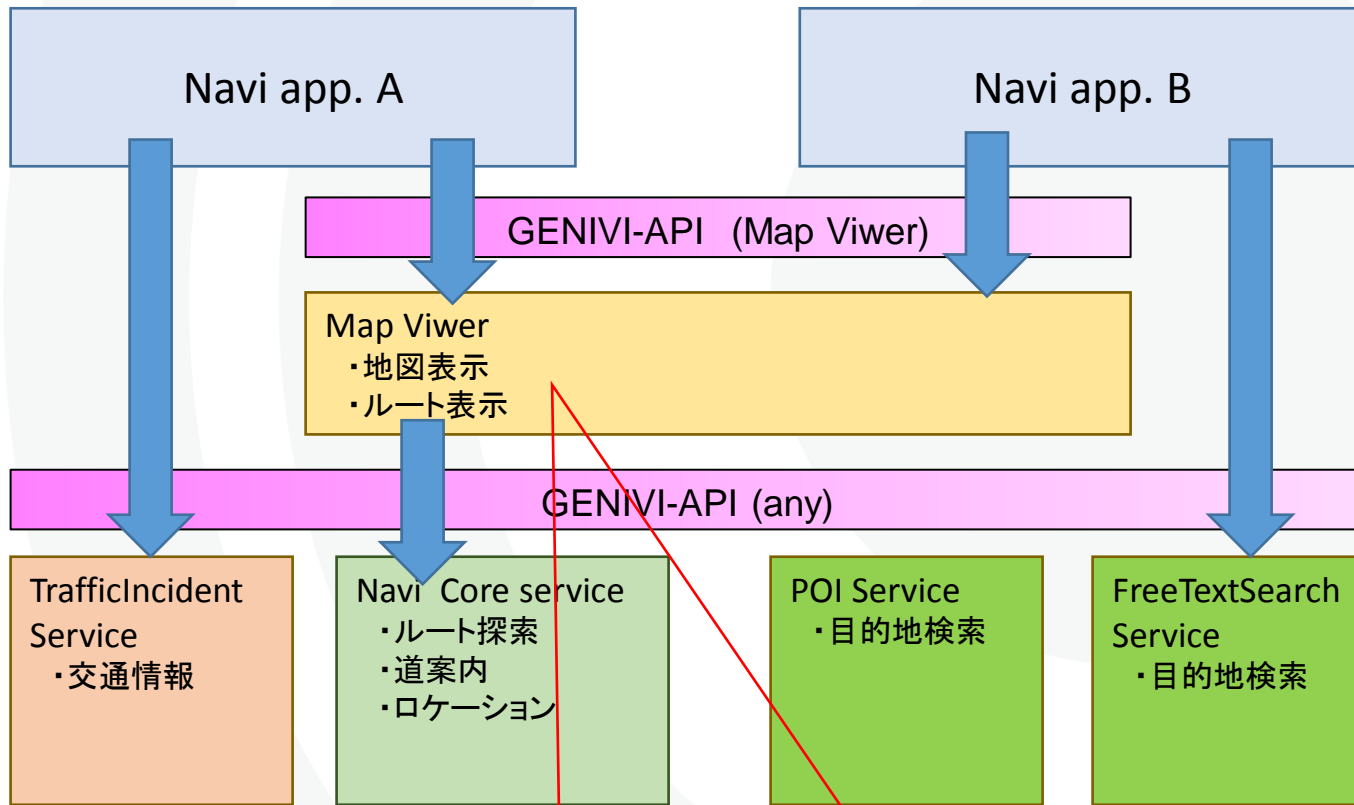
AGLはプロセス境界で、APIのセキュリティをかける方向に進んでいる。

AGLのHome Screenアーキテクチャは、複数プロセスで描画したサーフェスのコンポジットをサポートしていない。HMIと地図は、同じプロセスで1枚のサーフェスに描画する必要がある。

GENIVIのAPIはD-Bus上で実装できるようにかんがえられており、AGLとの矛盾はない。

GENIVIはLayer Managerによる表示管理を行っているため、異なるプロセスが描画したサーフェス(レイヤ)のコンポジットをサポートしている。MapViwerはこれを前提としているため、現状のAGLのアーキテクチャとは整合が取れていない。

AGLにおけるGENIVIアーキテクチャの課題



複数のナビアプリケーション(たとえば、通常の道案内ナビアプリと、地図上にポケスポットを表示するナビアプリ)を作りたいと思った場合、現状のアーキテクチャだとMap Viwerがそれぞれのアプリ用の地図を描画し、それぞれのアプリとLayerManagerでコンポジットしてもらえようコントロールする必要がある。
このモデルを目指す場合は、NaviEGとしてHomeScreenに要求していく必要がある。

今後に向けて

今後、前提とするアーキテクチャに関して議論を進めていきたいと考えています。

今回、そのたたきだいとして一例を挙げましたが、各社で課題出しを行いたいと思います。