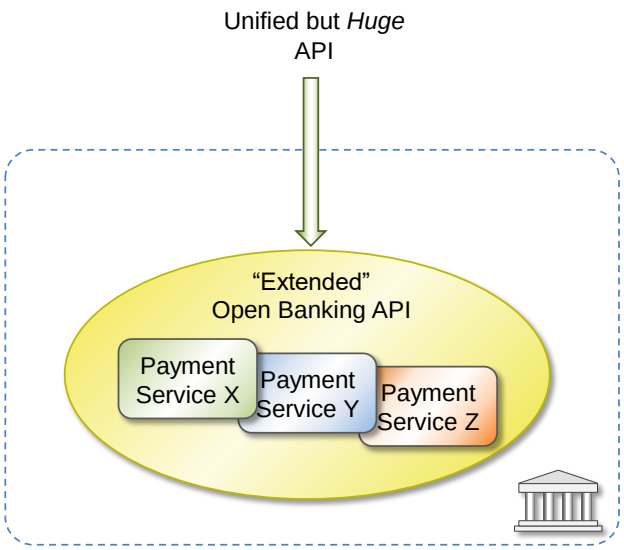
Embedded SCA versus Direct Mode

Feature	Embedded SCA	Direct Mode	Comment
Multiple Development Options	✘	☑	**Direct Mode** permits *anybody* to develop a payment service including keeping the design *private*.
Platform Independence	✘	☑	Since **Direct Mode** services are *external* to Open Banking APIs, they may build on any suitable platform.
Easy to Standardize	✘	☑	**Direct Mode** only requires a *single* and fully standardized change to work, while **Embedded SCA** is a framework leaving 99% to payment service developers to define.
"Sandbox" Compatible	✘	☑	**Direct Mode** payment systems can be developed, demoed, and marketed based on "Sandbox" implementations.
"Cloud" Compatible	✘	☑	**Direct Mode** permits multiple deployment models, including running payment services in the cloud.
Service Isolation	✘	☑	**Direct Mode** makes it logical running payment services on dedicated hosts making maintenance and upgrades more manageable than in *monolithic* Open Banking API implementations.
Support for popular P2P payment systems like "Swish"	✘	☑	**Direct Mode** services are *unrestricted* with respect to flow, security, and processes.
Cost Efficient	✘	☑	**Direct Mode** makes it technically possible creating a *single implementation/code base supporting all Banks.* This is probably a prerequisite for third party vendor support.
Concerted Updates	✘	☑	Updating **Embedded SCA** applications should be *quite challe*nging due to the diversity of Open Banking implementations.
Security / Authentication	☑	☑	In the **Direct Mode** security maintained by each of the connected services on their own. Each payment service must also authenticate to the Open Banking service. In the **Embedded SCA** model most of the security is confined to implementations-specific solutions inside of the Open Banking framework. *These methods should be comparable although it seems that it should be easier to audit software that is 1) used by more customers 2) is not a part of another framework.*

Embedded SCA



Direct Mode