Pre-Class Discovery Handout: Embedded Finance Business Models

Activity 1: Business Model Canvas Detective

Scenario: Pick ONE embedded-finance offering you have encountered — for example a rewards card issued through a consumer brand, a buy-now-pay-later checkout option, or a business-expense card inside a software tool. Fill in the canvas below by investigating who owns each block. Attribute every block either to the **brand** the customer sees, or to the **BaaS provider** that supplies the rails, or to the **sponsor bank** that holds the charter.

Canvas Element	Your Analysis (brand / rails / sponsor bank)
Value Proposition *What friction does this embedded offering remove for the customer?*	
Customer Segments *Who is the primary user and why do they engage through the brand?*	
Channels *How does the offering reach the user without a separate financial app?*	
Revenue Streams *What types of revenue flow, and between which parties (not amounts)?*	
Key Resources *What plumbing does each party contribute that the others cannot?*	

Q1: What is the single most important friction this embedded offering removes for the customer compared to visiting a bank directly?

Q2: Which entity does the customer believe they are transacting with — the brand, the rails provider, or the sponsor bank — and why does the answer matter for complaint handling?

Q3: If the BaaS provider switched sponsor banks tomorrow, what would change for the customer, and what would remain identical?

Activity 2: Unbundling Map

Scenario: A BaaS provider's API catalogue does not appear all at once. Each provider launches a wedge endpoint, then widens the surface. Match each of the following five BaaS providers on the left to the launch wedge service it is best known for — not its full current catalogue.

BaaS Provider		Launch Wedge Service
Solaris	____	Fully-licensed European credit-institution charter offered through an API
Unit	____	Deposit accounts exposed through a single-endpoint API for software teams
Treasury Prime	____	Network-based access to multiple community-bank charters
Marqeta	____	Modern card-issuer processing for programme managers
Synctera	____	Marketplace matching community banks with non-bank brands

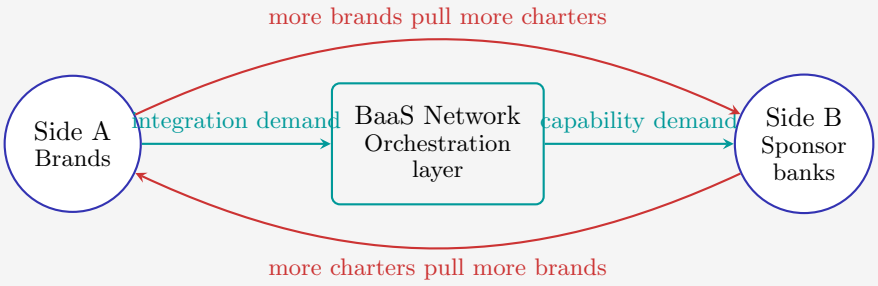
Q1: For each provider, describe in one sentence the single friction its wedge service removes for the brand that integrates it.

Q2: Which of these providers has most aggressively widened its API catalogue after the wedge? What was added, and in what order?

Q3: Why might a BaaS provider that launches with one wedge service eventually want to cover the full stack — and what does it risk by doing so?

Activity 3: The Platform Puzzle

Scenario: A BaaS network is a genuine two-sided platform. On one side are brands that want to embed financial products; on the other side are chartered banks that want non-interest fee income. The BaaS network is only useful when both sides are populated. Consider the diagram below.



Q1: Why does a BaaS network with more integrated sponsor banks become more attractive to brands, and vice versa? Name the specific cross-side mechanism.

Q2: Which side should a new BaaS network attract first, and why? How does the answer differ from a consumer payment network?

Q3: Once a BaaS network reaches critical mass, what specifically makes it hard for a later entrant to catch up?

Solutions

Activity 1: Business Model Canvas Detective

A1: Model answer for Solaris (from the perspective of a brand using Solaris rails): the most important friction removed is the need for the brand to apply for and maintain its own credit-institution licence. The embedded offering lets the brand issue regulated products — payment accounts, cards, lending — without operating as a bank. The friction disappears for the brand; for the customer, the friction removed is the need to open a separate bank account to engage with a financial feature of the brand.

A2: For the customer, the perceived counterparty is almost always the brand. Legally, however, the contractual counterparty is the sponsor bank that holds the charter; Solaris is the rails layer in between. The answer matters for complaint handling because conduct-of-business rules, deposit-insurance protection, and resolution regimes all attach to the sponsor bank, not to the brand. Customers who misidentify the counterparty can struggle to invoke the correct protections.

A3: If the BaaS provider switched sponsor banks, the customer would in most cases see continuity of the brand experience and minimal interruption. What would change behind the scenes: the regulatory counterparty, the deposit-insurance scheme covering the balances, the dispute-resolution escalation path, and the contractual framework governing the brand-BaaS-sponsor triangle. What would remain identical: the brand interface, the visible product features, and the customer's day-to-day flow.

Canvas elements (Solaris as the BaaS rails):

- **Value Proposition:** licensed credit-institution capability delivered through an API, letting brands issue regulated products without becoming banks.
- **Customer Segments:** primary — consumer brands and fintech firms seeking embedded payment, card, and deposit features; secondary — small banks and institutions that would otherwise build bespoke integrations.
- **Channels:** developer documentation, partnership sales teams, and integrator marketplaces. The end customer never sees these channels.
- **Revenue Streams:** platform access fees, per-account and per-transaction fees, and a share of interchange or spread negotiated with the sponsor-bank side.
- **Key Resources:** the credit-institution licence itself, the orchestration and ledger technology, the compliance and risk apparatus, and the operational team that supervises brand programmes.

Activity 2: Unbundling Map

A1: Solaris removes the friction of brands needing to obtain their own banking licence to issue regulated products. Unit removes the friction of integrating bank functionality through fragmented legacy APIs — it collapses many endpoints into one consistent developer surface. Treasury Prime removes the friction of brands being locked to a single sponsor bank by providing access to a network of charters. Marqeta removes the friction of modern card-programme launches, substituting a flexible issuer-processor for a legacy card-issuing stack. Synctera removes the matching friction between community banks looking for fee income and non-bank brands looking for sponsor capacity.

A2: Unit has most visibly widened its catalogue: it launched with deposit accounts, added card issuing and payment movement, and then layered lending and financing endpoints on top. The sequencing illustrates Christensen's unbundling-to-rebundling cycle applied at the API level: each earlier endpoint supplies the identity, ledger, and compliance plumbing that later endpoints require.

A3: Full-stack BaaS capture more per-customer lifetime revenue and embed themselves more deeply in each brand's product surface. The risk is complexity: each new endpoint consumes compliance

staff, sponsor-bank coordination capacity, and engineering attention that competitors with a narrower focus can outpace. A full-stack BaaS can also find itself in contractual competition with its own sponsor banks if its ambition expands beyond the hosted model.

Activity 3: The Platform Puzzle

A1: This is a **cross-side network effect** specific to BaaS. More integrated sponsor banks mean brands can choose a charter whose risk appetite, jurisdiction, and pricing match their needs, which makes the network more attractive to new brands. Simultaneously, more brands mean more fee-generating programmes for sponsor banks, which makes the network more attractive to additional banks. Each side's growth reinforces the other.

A2: A new BaaS network typically attracts the **sponsor-bank side first**, because a brand will not integrate with a network that cannot guarantee a charter match. This contrasts with a consumer payment network, which typically must seed the merchant side first to give consumers somewhere to transact. The BaaS network must first earn the trust and operational readiness of several sponsor banks before any brand integration becomes meaningful.

A3: A mature BaaS network accumulates a **structural moat** made of three layers: integration depth (brands have built product flows against specific APIs and cannot switch cheaply), regulatory capital (the compliance tooling and sponsor-bank partnerships built during scaling cannot be replicated in a single product cycle), and catalogue breadth (the incumbent already covers the endpoints a new entrant would need to match). A later entrant must reach minimum viability on all three simultaneously to attract the first bank and the first brand.

