

In-Class Exercise: Open Banking Business Models

Exercise 1: Structured Debate — “Is an Open-Banking Aggregator a Platform or a Utility?”

Format: Split into two teams. Each team prepares its position, then presents. After both sides speak, the class votes — but first read the debrief questions. For grounding, pick Plaid as the reference case, because its developer footprint spans both sides of the argument more visibly than most of its peers.

Team A — “Plaid Is a Platform”

Anchoring evidence: Plaid has wide developer adoption, charges for access, packages premium data products on top of raw connectivity, and benefits from cross-side effects as more applications integrate. Its Business Model Canvas reads like a two-sided platform business.

Team A: Plaid Is a Platform

Argument I

Argument II

Argument III

 Concession *Strongest argument AGAINST your position:*

 Closing *How you address the concession:*

Team B — “Plaid Is a Utility”

Anchoring evidence: Plaid’s core function is a commoditisable connector layer. Banks ultimately control the supply side by mandate, and large fintechs increasingly negotiate direct connectivity. If open banking becomes a regulated right rather than a service, Plaid looks more like an electricity meter than a platform.

Team B: Plaid Is a Utility

Argument I

Argument II

Argument III

 Concession *Strongest argument AGAINST your position:*

 Closing *How you address the concession:*

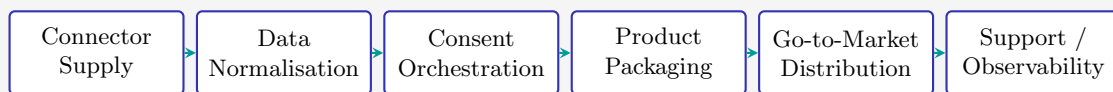
Debrief Questions

Q1: Does the answer — platform or utility — matter for how the aggregator should be regulated and priced? Why or why not?

- Q2:** Could the answer genuinely be “both, at the same time”? If so, what does that imply for the usefulness of the classification in the first place?
- Q3:** Name another financial-services company that blurs the platform-versus-utility boundary in the same way. What tension does the blurring create for its investors?

Exercise 2: Value Chain Mapping

Scenario: The open-banking value chain has six links. For each link, identify which of the five aggregators on the reference slate (Plaid, TrueLayer, Tink, Yapily, Salt Edge) most clearly owns the link, and describe what the aggregator is substituting for a direct bank-to-developer relationship at that link. Aggregators rarely own every link outright — be explicit when the aggregator is renting a link via a partner bank or SaaS vendor.



Value Chain Link	Aggregator Owning It	What It Substitutes For a Direct Bank API	Owned or Rented?	Is the Link a Moat or a Margin Cap?
Connector Supply				
Data Normalisation				
Consent Orchestration				
Product Packaging				
Go-to-Market Distribution				
Support / Observability				

Synthesis Questions

- Q1:** Which value-chain link creates the strongest moat for an open-banking aggregator, and which link most clearly caps its margin? Defend both answers with reference to engineering cost, switching cost, and regulatory positioning.
- Q2:** For at least two of the five aggregators on the reference slate, identify a link where the aggregator “rents” via a partner or a mandated supplier. What would it take for the aggregator to convert that rented link into an owned one — and should it?

Facilitator Solutions

Sample answers for instructor reference. These are illustrative; student reasoning may diverge and still be valid.

Exercise 1: Debate Sample Answers

Team A (Plaid Is a Platform) — sample arguments

Argument I. Plaid exhibits classic two-sided platform dynamics. Wider developer adoption makes the aggregator more attractive to banks that want efficient connectivity to many applications; wider bank coverage makes Plaid more attractive to developers who need portfolio-scale coverage. Those cross-side effects are durable because they are engineering-paced, not merely contractual.

Argument II. Plaid layers premium products on top of the raw connector: identity verification, income estimation, onboarding packages. Platform economics predicts exactly this pattern: commoditised base infrastructure monetised through higher-margin overlays that depend on the base but are not the base. A utility does not stack these overlays — it sells metered access at regulated rates.

Argument III. Plaid captures developer lock-in through SDK integration, data schema dependencies, and workflow habits. Switching an application's connector vendor requires re-integration effort that scales with the number of bank connections the application depends on. Platform-scale lock-in is a structural advantage a utility would not enjoy.

Concession. The strongest argument against Team A is that the underlying supply — bank APIs — is mandated by regulation, which means the aggregator's "platform" sits on top of a commoditised rail rather than an exclusive asset.

Closing. Platform status does not require exclusive supply; it requires cross-side effects, premium overlay monetisation, and switching-cost accumulation. Plaid demonstrates all three, regardless of how the raw supply is regulated.

Team B (Plaid Is a Utility) — sample arguments

Argument I. The core service Plaid provides — bank connectivity — is increasingly becoming a regulated right. Where supervisory frameworks mandate that banks expose standardised APIs, the value an aggregator can add on the connectivity layer compresses toward what a utility can extract: reliability, uptime, and scope. Utilities earn regulated returns, not platform returns.

Argument II. Large fintech applications increasingly negotiate direct connectivity deals with banks, bypassing the aggregator entirely for their highest-volume connections. This pattern mirrors what happens in telecommunications when large enterprise buyers acquire private fibre rather than riding public infrastructure. The trajectory of connector aggregation under mandate-driven standardisation is toward utility economics, not platform economics.

Argument III. Plaid's Cost Structure is dominated by connector-engineering maintenance — an operational cost that looks far more like grid maintenance than software platform investment. Platform economics rewards marginal cost approaching zero; utility economics rewards reliability at a managed marginal cost. Plaid's unit economics read utility-ward as the connector surface standardises.

Concession. The strongest argument against Team B is that Plaid's premium-data overlays (identity, income, onboarding) earn platform-like margins today, and those overlays could continue to compound even if the connector layer commoditises.

Closing. The connector layer is on a utility trajectory; only the overlays may remain platform-shaped. Classifying the entire business as a platform today masks the structural pressure on the base layer. Utility economics will dominate the long-run margin profile unless the overlay strategy outruns the mandate-driven standardisation.

Debrief Q1 — Regulation and pricing

The classification matters because it changes the prudential and competition-policy toolkit that applies. A platform is regulated for market-power effects, data-portability obligations, and fair-access terms; a utility is regulated for regulated returns, universal-service obligations, and price caps. Treating Plaid as a platform allows pricing discretion and premium overlays; treating it as a utility pushes toward cost-plus pricing and tariff regulation. In practice, most jurisdictions have not yet decided, which is why aggregators operate in a hybrid regime and investors must forecast which end of the hybrid the regulator will land on.

Debrief Q2 — “Both” as an answer

The answer can genuinely be “both at the same time.” The connector layer has utility characteristics — commoditisable, mandate-adjacent, reliability-priced — while the overlay layer has platform characteristics — cross-side effects, premium products, switching costs. If both labels are correct simultaneously, the classification becomes functional rather than institutional: regulate the connector layer as a utility while letting the overlay layer compete as a platform. That framing is already visible in supervisory proposals that distinguish between access APIs (mandated, standardised) and premium APIs (commercial, negotiated).

Debrief Q3 — Cross-sector blurring example

Cloud infrastructure providers such as Amazon Web Services or Microsoft Azure provide a close parallel in the technology sector: they offer commoditisable compute and storage that could be treated as utility infrastructure, alongside premium services (managed databases, machine-learning tooling, industry-specific verticals) that look like platform products. Investors and regulators have struggled to classify them: competition authorities probe for platform-style self-preferencing, while enterprise buyers increasingly push for utility-style portability. The tension is directly analogous to Plaid’s, and it creates uncertainty for investors about which peer set should anchor valuation multiples.

Exercise 2: Value-Chain Mapping Sample Answers

Value Chain Link	Aggregator Owning It	What It Substitutes for a Direct Bank API	Owned or Rented?	Moat or Margin Cap?
Connector Supply	Plaid (portfolio-scale connector library across North American banks)	One-by-one bank integrations; substitutes engineering cost for individual partnership paperwork	Rented (mandated or contractual supply from banks)	Margin Cap (regulated supply ceiling)
Data Normalisation	Tink (consistent data schema across Nordic and European banks)	Heterogeneous bank-specific formats; substitutes a normalised schema for bespoke parsing	Owned	Moat (engineering effort compounds; hard to replicate)
Consent Orchestration	TrueLayer (unified consent surface across data and payments)	Separate consent flows per bank per product; substitutes a single branded surface	Owned	Moat (switching-cost barrier for developers)
Product Packaging	Plaid (identity, income, onboarding overlays on top of raw data)	Raw-data integration with no value-added layer; substitutes premium packages sold as products	Owned	Moat (premium-data margins outlast connector commoditisation)
Go-to-Market Distribution	Yapily (enterprise-buyer-focused sales without end-user competition)	Bank-led outreach that pits the bank against downstream fintechs; substitutes an aligned vendor posture	Owned	Moat (enterprise-trust positioning is hard to reproduce)
Support / Observability	Salt Edge (cross-region uptime monitoring and schema-drift detection)	Per-bank operational support; substitutes shared SaaS-grade observability	SaaS (shared tenant)	Margin Cap (infrastructure shared across buyers)

Synthesis Question 1 Sample Answer

The link that creates the strongest moat for an open-banking aggregator is Data Normalisation. Consistent data schemas across heterogeneous banks are compounding engineering assets: every schema drift absorbed, every edge case handled, every latency variance smoothed becomes knowledge that a later entrant must re-learn from scratch. Developers building on the aggregator accumulate dependencies on those schemas, raising switching costs sharply. Regulatory barriers are light on this link (no licence is required to normalise data), but the engineering moat is the dominant barrier. The link most likely to cap margin is Connector Supply, especially in mandated-supply regimes. The supplier is a licensed bank that neither competes commercially nor sells access, which means the aggregator cannot capture pricing power on the supply side. Converting Connector Supply into an owned link would require the aggregator to become a licensed bank itself — structurally impractical — so the rented posture is the rational steady-state and the margin ceiling follows.

Synthesis Question 2 Sample Answer

Plaid rents connector supply from banks across its region; converting that rented link into an owned one is not structurally possible without acquiring the banks themselves, which is neither legal nor economically sensible. What Plaid can do is deepen the owned overlay layers so the rented link's margin ceiling becomes less binding — identity, income, and onboarding products

that sit above the raw connector. Salt Edge rents Support and Observability as shared SaaS infrastructure; converting it to an owned link would require Salt Edge to build dedicated observability tooling for each enterprise buyer, which would raise cost without materially raising revenue. The general pattern for aggregators is that only some rented links can be converted into owned ones; the rest represent structural limits of the business model, and forcing conversion would destroy the cost advantage that let the aggregator scale in the first place.