

Implementing Toast Loyalty Points on the Blockchain

Author: Jason L. Lind – jason@groundbreaker.solutions

Date: September 10, 2025

Executive Summary

This proposal outlines a plan for Toast, Inc. to evolve its restaurant loyalty program by tokenizing loyalty points on a blockchain and enabling their transfer or trade between customers and participating restaurants (“tenants”). The goal is to enhance customer engagement and cross-brand loyalty while maintaining compliance and accounting integrity. We recommend issuing loyalty points as blockchain tokens (using the ERC-1155 standard) to provide transparency, security, and interoperability across the Toast network. Key considerations include ensuring regulatory compliance (treating tokens as utility points, not securities), leveraging smart contracts for controlled issuance/redemption, and integrating seamlessly with Toast’s POS and loyalty systems. We propose a phased rollout—starting with a pilot and progressing to full network deployment—with close attention to user experience, merchant economics, and risk governance. Through this initiative, Toast can pioneer a multi-restaurant loyalty marketplace, driving value for customers (points with broader usability and tangible value) and for tenants (attracting new customers and insights from a shared loyalty ecosystem), all while positioning Toast as an innovator in restaurant tech.

1. Background

Toast’s current loyalty program allows individual restaurants to reward customers with points for purchases, redeemable for discounts or freebies on future visits. These traditional loyalty points are siloed to each restaurant brand and are typically non-transferable and have no cash value until redeemed[1]. Industry-wide, such fragmentation limits the usefulness of loyalty programs—customers juggle multiple separate point currencies, many of which go unused and expire[2][3]. Studies indicate ~70% of loyalty program members do not redeem their points, often due to limited redemption options and lack of interoperability[3]. Forward-looking businesses are exploring coalition programs and tokenized loyalty assets to address these pain points. For example, partnership-based loyalty networks (airline alliances, retail coalitions) have grown, allowing points earned in one business to be spent at another[4][5]. Blockchain technology has emerged as a promising infrastructure for such networks by providing a secure, shared ledger where multiple companies can co-manage a loyalty currency without a single centralized broker[6]. Notably, major firms like JPMorgan Chase are even experimenting with turning loyalty points into **tradeable assets**[7], highlighting a trend to increase liquidity and customer value from points. Starbucks, too, has expanded its successful Rewards program with blockchain-based extensions (e.g. NFT collectibles via “Starbucks Odyssey”) to boost engagement. These developments underscore an opportunity for Toast to

leapfrog legacy loyalty limitations. By implementing blockchain-based loyalty tokens, Toast can unify or connect disparate restaurant loyalty programs on its platform, making points **interchangeable across participating brands** and even **tradeable peer-to-peer**, all while leveraging the transparency and security benefits of distributed ledger technology[8][9]. In short, the evolving loyalty landscape and web3 tools provide Toast a timely chance to enhance its loyalty offering in a way that delights customers (more ways to use rewards), benefits restaurants (broader reach and data), and differentiates Toast in a competitive POS market.

2. Strategic Rationale

Implementing blockchain-based, tradeable Toast loyalty points supports several strategic objectives:

- **Enhance Customer Engagement & Loyalty:** By increasing the flexibility and utility of points, customers have a stronger incentive to join and actively use the loyalty program. When points can be used at multiple venues or traded with others, their perceived value rises[9]. This “liquidity” means guests are more likely to earn and redeem rewards (rather than letting them expire), leading to more frequent return visits[10]. A token that holds real exchangeable value provides an aspirational element to the loyalty program, tapping into the appeal of digital assets and even speculation (customers may hold or trade points if they expect them to appreciate in value with demand)[11][12]. Overall, a more engaging loyalty experience can increase customer lifetime value and satisfaction.
- **Differentiation & Network Effects:** This innovation would position Toast as a pioneer among restaurant tech providers. Few if any mainstream POS companies offer a blockchain-based loyalty marketplace. By being first-to-market, Toast can attract forward-thinking restaurant brands and create a network effect: the more merchants and customers participating in the tokenized loyalty network, the more valuable the network becomes for all. Customers will choose Toast-powered restaurants to earn/use universal points, and restaurants will want to join to gain access to this larger loyalty ecosystem. This strategic differentiation can strengthen Toast’s competitive moat in the POS industry.
- **Cross-Promotion and New Revenue Streams:** A tradeable loyalty token opens possibilities for cross-restaurant promotions and partnerships. For example, a group of local eateries on Toast could agree to accept each other’s points or run joint campaigns, effectively forming a coalition loyalty program. Blockchain’s transparency and security make it easier to settle value among partners without a single company controlling the ledger[6]. Toast could facilitate a **point exchange** where, say, “Coffee Shop A” points convert to “Bakery B” points, driving referrals between businesses. Additionally, a marketplace for points could allow **Toast to earn transaction fees or commission** on point trades or redemptions. There is precedent for this approach: JPMorgan’s partnership to create loyalty point

financial assets implies a market for trading point portfolios[7]. Toast could similarly capture value by enabling and managing these exchanges (subject to legal considerations). Furthermore, if restaurants wish to buy additional points (for instance, to distribute as promo rewards), Toast could sell or auction tokens to them, creating a new revenue line.

- **Transparency, Security, and Trust:** Moving loyalty transactions to a blockchain ledger increases transparency for all stakeholders. Customers can verify their point balances and transaction history on an immutable ledger, fostering trust that their rewards are secure and won't arbitrarily change or disappear[13][14]. Restaurants in a multi-tenant network can trust the integrity of point issuance and redemption data, since blockchain entries are tamper-proof and auditable by all partners[6]. This co-ownership of data is particularly useful in a coalition program—no single entity can secretly inflate or deflate the currency or misreport redemptions[15]. Enhanced security (cryptographic controls) also reduces the risk of fraud (e.g. fake or duplicate loyalty accounts) and data breaches of centralized databases[16][17]. All these factors can strengthen the loyalty program's credibility and longevity.
- **Future-Proofing and Technological Leadership:** Embracing blockchain and tokenization aligns with broader trends in digital assets and Web3 consumer engagement. Many retail and hospitality brands are experimenting with tokens and NFTs to modernize their loyalty and membership offerings (for instance, airline miles on blockchain, or NFT-based VIP clubs)[18][19]. Visa recently unveiled a Web3 loyalty platform, indicating growing mainstream acceptance of the technology in loyalty contexts[20]. By acting now, Toast can build internal expertise and IP in blockchain applications, positioning itself at the forefront of this trend. This not only has marketing value but prepares Toast to integrate with other emerging digital ecosystems (e.g. potential future where loyalty tokens from different platforms interoperate, or integration with cryptocurrency wallets). The initiative can also unlock secondary strategic benefits such as loyalty tokens being used for other purposes down the line (e.g. token-gated experiences, or even as a currency in Toast's broader product suite if applicable).

In summary, tokenizing Toast loyalty points addresses current program limitations (inflexibility, low engagement) and yields competitive and financial advantages. It aligns with an industry shift toward treating loyalty points as assets with real value[7], and supports Toast's mission of empowering restaurants with cutting-edge technology. The strategic upside must be balanced with careful execution to manage compliance, user experience, and risk—areas detailed below.

3. Compliance & Accounting Considerations

Implementing tradeable loyalty tokens requires diligence to ensure legal and accounting requirements are met. Key considerations include:

- **Regulatory Classification:** We must ensure Toast loyalty tokens are clearly characterized as **promotional loyalty points (utility tokens)** rather than securities or e-money. The tokens should not confer any ownership stake, fixed redemption for cash, or profit expectation beyond their use for discounts, which helps avoid securities implications. They will function as a right to receive a discounted or free product (similar to a coupon or stored-value gift card) and carry terms that they are **not redeemable for cash** (except as required by law in certain jurisdictions). Many loyalty programs today specify points have no cash value and are non-transferable^[1]; by explicitly allowing transfer/trade, we change one aspect, but we will retain that they are only officially redeemable for goods/services within the network. This nuance is important: if users trade tokens P2P, any cash exchanged is outside the platform's official scope (similar to how people can sell gift cards to each other). We will include clear **Terms & Conditions** that outline permissible use of tokens and disclaim any monetary value guarantee by Toast. Engaging legal counsel early is critical to navigate varying laws: for example, in the U.S., state gift certificate laws might apply if points are widely transferable or don't expire, and **unclaimed property laws** could apply to unredeemed points after a certain period. By structuring the tokens with an expiration "vintage" and keeping them as promotional credits, we aim to fall under loyalty program exemptions rather than stored-value regulations. However, because trading introduces quasi-liquidity, we will monitor guidance from regulators. (Notably, the SEC and other bodies have generally not treated typical loyalty points as securities, and our design will mirror typical loyalty use-cases to stay in that safe zone.)
- **AML/KYC:** Traditional loyalty programs rarely require KYC (Know Your Customer verification) since points are only used for purchases and have limited scope. Introducing transferability raises the possibility (albeit likely low) of abuse, such as using points to launder small sums of money. We will assess thresholds above which identity verification might be required for token transactions (for instance, if a user is trading a large volume of points with others). Given that restaurant loyalty points will have relatively modest value (e.g. 1 point = \$0.01 or \$0.10), the risk of large-scale money laundering is low. Nevertheless, to be prudent, the platform can **monitor unusual activity** (such as bulk purchases and transfers of points that don't correlate to actual dining activity). We can impose limits on P2P transfers or require transfers to occur within the Toast app ecosystem (not arbitrary blockchain addresses) to maintain oversight. Using a **permissioned wallet system** (where each user's wallet is linked to their verified Toast account) can help ensure we know who is transacting. If needed, we can geo-fence or restrict transfer features in regions with stricter e-money laws until compliance is confirmed. Overall,

compliance will be managed in a way that does not overly burden the casual user but protects the company from facilitation of illicit activity.

- **Data Privacy:** Because blockchain transactions are transparent, we must ensure no personal data is written to the public ledger. Our design will use anonymous wallet addresses (or hashed user IDs) on-chain. The mapping of those addresses to actual user identities will be kept in Toast's secure databases (off-chain), which is similar to how current loyalty accounts are managed. This ensures compliance with privacy laws (GDPR, CCPA) since personal information isn't exposed on a public ledger. We'll also clearly update our privacy policy to inform users how data is handled in this new system (for instance, disclosing that their loyalty balances and redemptions are recorded on a blockchain but in an anonymized manner).
- **Accounting (Revenue Recognition & Liability):** Under accounting standards (IFRS 15 and ASC 606), loyalty points are treated as a **separate performance obligation** – essentially deferred revenue – since the customer effectively prepays for a future benefit when earning points[21][22]. This will remain the case with tokenized points. For each sale where points are awarded, a portion of the sale's revenue will be deferred as a liability ("unredeemed loyalty points") based on the standalone value of the points and expected redemption rates. The blockchain implementation does not fundamentally change this, but it adds transparency and potentially impacts estimates (because if points are tradeable, redemption rates might increase toward 100% as users can transfer unwanted points to someone who will use them). For example, if historically a restaurant estimated 10% of points expire unused, that breakage may drop if a secondary market lets customers sell points they won't use. Accounting will need to adjust estimates of point utilization. The liability for issued tokens will only be removed when points are redeemed (or officially expired). **Each merchant** will still be the one ultimately funding the redemption of points they issue – i.e. providing free/discounted product when points are used. If we introduce a **cross-merchant redemption** (universal points), we must set up inter-company settlement so that the cost of redemption is borne by the issuing entity or shared according to agreements. One model is analogous to airline alliances: points have a known monetary value and when used at another partner, the issuing company compensates the redeeming company. Toast could operate a clearinghouse that monthly or quarterly tallies how many of Restaurant A's tokens were redeemed at Restaurant B and facilitate the net settlement between A and B. In accounting terms, if Toast itself issues a unified currency, Toast might hold the deferred revenue liability and relieve it when points are redeemed by paying the merchant in cash or credits. This scenario (Toast as the "bank" of loyalty points) could trigger revenue recognition for Toast as an agent or principal of the loyalty arrangement[23][24]. We will likely keep it simpler initially: each restaurant's points remain their liability, and cross-use is achieved through trading rather than direct cross-redemption (meaning the customer who wants to use points at another restaurant trades for that restaurant's points, transferring the liability with the

token). This way, each merchant accounts for their own points only. We will provide tools or guidance to merchants' finance teams on how to account for token transactions, which can be treated similarly to current points but with possibly lower breakage assumptions. All on-chain transactions will have off-chain records in Toast's system to support accounting audits, and the blockchain ledger can serve as an additional audit trail of issuance and burn events.

- **Tax Implications:** Generally, loyalty point issuance is not a taxable event for the customer (it's seen as a reduction of the sales price or a marketing incentive), and redemption is essentially giving away product (which might be recorded as a discount or promotion expense for the merchant). Trading of points between customers introduces a new wrinkle: if a customer sells points to another for cash (outside the platform's official functionality), that could be considered personal income for the seller in some jurisdictions. However, these amounts are likely small and below taxable thresholds, and Toast is not a party to those cash transactions. We will clarify in terms that users are responsible for any implications of trading tokens P2P. As for merchants, if they acquire tokens (say a merchant buys tokens on the marketplace to distribute in a promotion), that purchase might be treated like buying gift cards (an asset that will be expensed when given out). We will advise participating tenants to consult their accountants on treating any token purchases or sales. Toast itself could incur tax obligations if it earns fees on the marketplace or if it holds a reserve of tokens that appreciate, but the plan is not to hold tokens for speculative purposes, only to administer the program.
- **Legal Compliance (Consumer Protection):** We will ensure the program complies with consumer protection rules for loyalty programs. For example, if we impose expiration on tokens (vintages), we'll follow any laws requiring adequate disclosure or minimum validity periods. Some jurisdictions mandate that loyalty points or gift certificates cannot expire or must be redeemable for cash above certain values – we will either avoid expiry or allow conversion if legally required. The blockchain design actually *facilitates* compliance here, as it can enforce expiry automatically by disallowing redemptions after a cutoff date for a given token vintage. If a jurisdiction (like certain U.S. states) requires offering cash back once a certain amount of points accrues, we could programmatically enable that for those users or simply handle it off-chain through customer support. All of these conditions will be codified in the smart contract logic where possible, ensuring consistent application of rules.

In summary, while the introduction of tradeable loyalty tokens is novel, we will design the system within the existing frameworks for loyalty programs. By proactively addressing the above compliance and accounting factors – and possibly obtaining a third-party legal opinion or regulatory sandbox approval – Toast can mitigate risks and confidently roll out the tokenized loyalty system.

4. Token Design (ERC-1155, Vintages, Transfer Models)

Token Standard – ERC-1155: We propose to implement Toast loyalty points using the **ERC-1155 multi-token standard** on an Ethereum-compatible blockchain. ERC-1155 is well-suited for this use case because it supports creating **multiple token types within one smart contract** (unlike ERC-20 which is a single fungible token, or ERC-721 which is single-token non-fungible)[25][26]. This means we can manage different categories of loyalty points in one contract – for example, points for different restaurant brands, or different “vintages” – without deploying numerous separate contracts. It is efficient and reduces gas costs by allowing batch operations (e.g., a user could redeem multiple types of points in one transaction)[27][28]. Additionally, ERC-1155 provides safety features like preventing tokens from being stuck in contracts and a unified interface for wallets and marketplaces to interact with all token types[29][30]. Many blockchain loyalty and reward implementations have gravitated to ERC-1155 for these reasons[31]. By using this standard, Toast’s tokens will be compatible with a wide range of wallets, exchanges, and tools, giving us flexibility and users more control. However, we will likely **deploy on a Layer-2 network or sidechain** (such as Polygon or an Optimistic rollup) for low fees and scalability, while still leveraging the ERC-1155 standard. The contract can be upgradeable (proxy pattern) to allow future enhancements, though governance around upgrades will be defined to maintain trust.

Token Types and “Vintages”: Each loyalty token will represent a specific category of points. We envision two main axes for categorization: - **By Merchant (Tenant):** In the initial phase, it may make sense that each restaurant (or each restaurant group/brand) has its own token type under the ERC-1155 contract. For example, “Joe’s Diner Loyalty Token” might be token ID 1001, “Pasta Palace Loyalty Token” is ID 1002, etc. Points issued by a given merchant are of one type and are redeemable at that merchant. The multi-token contract allows this easily – it can assign a unique token ID to each merchant as they join the program[32]. The contract will restrict **minting rights** so that only the merchant (or Toast on behalf of that merchant) can issue their specific token, preventing any other party from creating counterfeit points[32][33]. This design essentially mirrors the status quo (each merchant’s points separate) but now on a shared ledger. Customers could hold many types of tokens in one wallet representing their points across different restaurants.

- **By Vintage (Expiration/Series):** We can incorporate “vintages” by segmenting tokens based on issuance date or campaign. For instance, a common practice in loyalty programs is that points expire after a certain time (e.g., 12 or 24 months after earning). We could implement this by creating a new token ID for each calendar year (or quarter) of issuance. Example: “Joe’s Diner 2025 Points” (token ID 1001-2025) and “Joe’s Diner 2026 Points” (ID 1001-2026) could be treated separately. Points earned in 2025 would automatically expire end of 2026, for example, and that expiration can be enforced by the contract burning the 2025 token series after that date, or simply refusing redemption transactions on expired tokens. This vintage approach is inspired by systems like carbon credits and some blockchain loyalty research where each batch/year of points is tracked distinctly for lifecycle

management[34]. Using ERC-1155, handling multiple vintages is straightforward since it thrives with many token IDs. Vintages also allow for possible differentiation – perhaps points earned during a special promotion have a different ID and rules (e.g., a “promotional token” that might only be usable at certain times or with certain partners). For the baseline design, vintages = expiration grouping. This gives clarity to users (the app can show “You have 100 points expiring Dec 2025”) and ensures liability doesn’t persist indefinitely.

In summary, the ERC-1155 contract might have $Token\ ID = MerchantID * 1000 + Year$. The contract will map these to metadata like merchant name and expiry date. We will maintain an off-chain registry for ease of lookup as well.

Value and Fungibility: All tokens of the same type (same merchant & vintage) will be fungible (interchangeable) and carry the same face value (e.g., 1 token = \$0.01 in rewards at the issuing merchant, value to be defined by program rules). Different token types are not directly fungible with each other (1 point of Restaurant A is not necessarily equal to 1 point of Restaurant B in market value), which is why a marketplace or swap mechanism is needed for exchange (discussed below). We may also introduce a **universal Toast token** in the future (a “Toast Coalition Point”) that all merchants accept, effectively pegged to a currency value. This would simplify things for users but requires merchants to agree on settlement value. One hybrid approach is to use the universal token as an intermediary for swaps (see **Marketplace Design** in Section 9). Initially, however, focusing on separate tokens per merchant with trading is simpler and avoids cross-merchant liability complexities.

Transfer Models: We outline how tokens can be transferred between wallets and what constraints are applied, defining three models:

- **Closed (Current Model – No Transfer):** By default, loyalty points are non-transferable (other than back to the issuer on redemption). This is how current systems work and is the baseline we are changing. We mention it only as a reference; our aim is to move to more open models.
- **Controlled P2P Transfer (Phase 1 Model):** We propose to launch with a **peer-to-peer transfer feature within the Toast platform**. This means users can **gift or send points to another user** intentionally, but with some platform mediation. For example, a customer could open their Toast app, choose to send 50 “Pizza Place” tokens to a friend. The app (or web portal) would initiate an on-chain transfer from the sender’s wallet to the recipient’s wallet. This is a straightforward blockchain transaction, but by keeping it in-app we can ensure the user experience is smooth (users don’t need to copy long addresses) and we can log the reason (gift, etc.). Controlled P2P also allows us to implement any needed checks (e.g., perhaps we require that both users have verified accounts to prevent abuse, or limit daily transfer volume). Initially, we might restrict transfers to **zero-price transfers** (gifting) and disallow direct point-for-money trades on the platform until we gauge

user behavior. Even with only gifting, this adds value – families or friend groups can pool points to reach a reward, for instance.

- **Open Trading via Marketplace (Phase 2 Model):** In the full vision, users and even tenants can **trade tokens openly**. To facilitate this, Toast will implement a built-in **marketplace contract** where offers to swap one type of token for another (or for a universal token or stablecoin) can be matched. The smart contract can act as an escrow: User A wants to trade 100 of Restaurant X’s tokens and is asking for at least 200 of Restaurant Y’s tokens in return; User B has Restaurant Y tokens and wants X’s tokens – the contract swaps them if criteria match. This could also be done via an order-book or automated market maker model. Another approach is using a **universal loyalty token** as a bridge currency^{[35][36]}: e.g., a user converts their specific points to a generic Toast token at a floating rate, and then converts that to another specific point type. That requires maintaining exchange rates either via market-driven prices or pegging (perhaps Toast token = 1 cent always, and each point type trades around that based on perceived value). The exact mechanism will be determined after studying expected volumes – a simple auction listing board might suffice initially, whereas high volume might warrant an AMM (Automated Market Maker) pool for each pair. Regardless of mechanism, the tokens themselves are inherently transferable on-chain to any address. If a sophisticated user wanted to, they could transfer tokens to a personal external wallet or trade them on third-party decentralized exchanges. We will not prevent that because doing so on an open blockchain is hard (we could put transfer restrictions, but that undermines user control and market formation). Instead, our **transfer model will be mostly open** but with guardrails: the smart contract can include an “**allow list**” of approved marketplace contracts to facilitate trading in a safer, tracked manner. For instance, we might enforce that any transfer between two user wallets must be either (a) a gift (no compensation, which we can’t fully know on-chain, but assume if they use our UI it’s a gift), or (b) executed via our official marketplace contract (which can log trade details). This way, if regulation ever required us to shut off trading, we could disable that contract and still allow gifting and redemption. Technically, ERC-1155 supports a feature called `safeTransfer` and hooks that could be used to restrict or log certain transfers^[30]. We will weigh using those to maintain some oversight. However, it’s important to maintain a degree of **openness** to gain the benefits (user trust and freedom, market price discovery, etc.).
- **Merchant (Tenant) Transfers:** “Tradeable among tenants” implies merchants themselves might want to exchange or acquire points. In our design, a merchant (like any user) can hold a wallet and thus hold tokens (including those of other merchants). One scenario: a merchant could buy another merchant’s points in the market – perhaps to run a promotion (“We’ll accept 100 of Bakery B’s points for a free coffee at our cafe – we acquired those points so Bakery B will effectively reimburse us when they’re redeemed”). This kind of B2B exchange could spur partnerships. Another scenario is a merchant deciding to **buy back** their own points

from the market to reduce liability (if their points are trading at a discount, they might buy them and then possibly retire them, similar to a company buying back gift cards). Our system will allow merchants to use the marketplace like any user. Additionally, the smart contract can allow an **admin burn** or “buyback” function if needed, although typically burning would only occur upon redemption or expiry. Governance (section 9) will outline how much control a merchant has versus Toast for such actions.

Token Distribution and Minting Controls: Only authorized entities will mint tokens. Toast (or specifically, the Loyalty Manager service) will have a master privilege to mint any token type, which it will use when a transaction at a restaurant triggers point issuance. We can also assign each merchant’s corporate account the ability to mint their own token type, as noted from research where each retailer had an owner key for their token in an ERC-1155 contract[32]. This ensures no one can maliciously create points. The rate of issuance (e.g., 10 points per \$1 spent or whatever the merchant sets) will be configurable in the loyalty software but enforced off-chain by the Toast middleware when calling the mint function. The smart contract could, if we want extra safety, implement a cap or rate limit per merchant (for instance, no more than X points per day without special override) to prevent errors causing runaway inflation of points.

On the redemption side, tokens will be **burned** when redeemed for rewards (or transferred to a null address or a dedicated contract address that represents “spent” points). Alternatively, we might transfer redeemed tokens into the issuing merchant’s own address as a record of fulfillment, then allow merchants to periodically burn them to reduce total supply. The decision between burn vs. recycle depends on how we want to track outstanding liabilities; burning is simpler and shows a clear reduction in total supply (which anyone can audit on-chain to see that liabilities were honored).

Value per Token: We will set a standard valuation (likely 1 token = \$0.01, if we keep current typical loyalty value). Merchants could choose to structure differently (some might give more points per dollar and require more points to redeem, but effectively all that matters is the monetary value). Keeping a base unit consistent (like 100 points = \$1 across the board) might make trading easier for users to understand, though the marketplace will anyway determine relative value. If Merchant A’s points are easier to earn (say they give 20 points per \$ and redeem at 100=\$1, effectively a 20% reward rate) versus Merchant B gives 5 points per \$ (5% reward rate), the market might value A’s points lower individually since they are more inflated. However, since each is only usable at its own merchant, the value to a user is fundamentally anchored by what it gets you at that merchant. These dynamics will be interesting and will encourage merchants to calibrate their earn/burn rates competitively. Toast can provide guidance to ensure one program doesn’t wildly out of sync with others unless intentionally.

Inactive/Expired Token Handling: With vintages, each expired series can be globally burned via an administrative function once past expiration, to simplify the token ledger and clearly extinguish the liability (after notifying stakeholders). If a merchant discontinues

using Toast or leaves the program, we'd have to decide how to handle their outstanding tokens – possibly freezing transfers and redemptions for that token and offering users a period to redeem at that restaurant (if it's still operating) or maybe at others as a courtesy. This scenario would be covered in merchant agreements (they might have to escrow some funds for outstanding points if leaving).

In conclusion, the token design will be **modular and flexible**, leveraging ERC-1155 for multi-token efficiency, using vintages to manage expirations, and implementing transfer controls that evolve from cautious (in-platform gifting) to open (market trading) as the ecosystem matures. The smart contracts will be thoroughly audited for security, and follow best practices (using OpenZeppelin libraries, etc.). By citing existing patterns (such as owner-restricted minting[32] and meta-transaction support[37] from academic prototypes), we ensure a robust design grounded in proven concepts.

5. System Architecture (Toast Integration, Smart Contracts, Wallets)

Implementing blockchain loyalty will require a robust architecture that links Toast's existing systems (POS terminals, loyalty database, merchant interfaces) with the blockchain network. Below we describe the proposed architecture and its components, and illustrate how data will flow from a restaurant transaction to the blockchain and back.

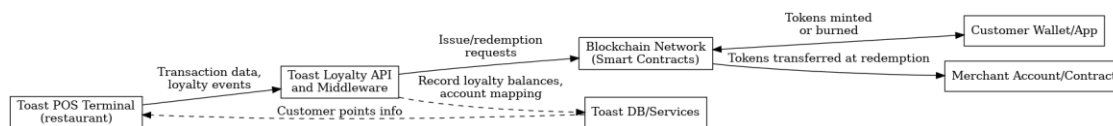


Figure: Proposed system architecture integrating Toast's POS and Loyalty systems with a blockchain layer (smart contracts and wallets). The POS transaction triggers loyalty events via Toast's backend, which interfaces with the blockchain to mint or burn tokens. Off-chain databases and services remain in place for user account management, fast lookup of balances, and ensuring smooth POS operations.

At a high level, the architecture consists of the following elements:

- **Toast POS and Backend:** Restaurants will continue to use their Toast POS terminals as usual. When a customer identifies themselves (e.g. through a phone number, loyalty QR code, or linked credit card) during a transaction, the POS will send the transaction details to Toast's cloud backend. Toast's existing Loyalty API (or an enhanced version) will capture events like points earning or redemption requests from the POS[38]. This backend will now be augmented with blockchain integration logic.
- **Loyalty Middleware Service:** We will introduce a middleware service in Toast's cloud that acts as an intermediary between Toast's traditional systems and the blockchain network. This service will handle tasks such as: constructing blockchain transactions (mint, transfer, burn calls to smart contracts), signing them with the appropriate keys, and broadcasting to the network. It will also listen for events from the blockchain (like a confirmation that tokens were transferred) and update

Toast's off-chain database accordingly. In essence, this service encapsulates the blockchain complexity and exposes simpler APIs to the rest of Toast's systems. For example, after a sale, the POS or loyalty engine can call `awardPoints(userID, merchantID, points)` on this service; the service will then execute the ERC-1155 `mint(merchantTokenID, userWallet, amount)` function on-chain. Likewise for redemptions.

- **Blockchain Network:** We propose using an Ethereum-compatible blockchain for the token ledger – this could be a **permissioned consortium chain** (operated by Toast and perhaps some anchor partners) or a **Layer-2 network** on Ethereum. A permissioned network (e.g., using Hyperledger Besu or an EVM sidechain) could offer more control (only Toast/merchants as validators, low or no gas fees in native currency) at the expense of decentralization. An open L2 (like Polygon, Arbitrum, etc.) offers more security and user autonomy but introduces gas fees and some complexity (however, those can be abstracted with meta-transactions). In either case, the smart contracts (ERC-1155 token contract and possibly marketplace contract) reside on this network. We will ensure the network choice supports **fast confirmation times** (a few seconds or less) to avoid delaying POS interactions.
- **Smart Contracts:** The primary contract is the **Toast Loyalty Token ERC-1155 contract** managing all point tokens. It will include functions like `mint(merchantTokenId, toAddress, amount)`, `burn(fromAddress, merchantTokenId, amount)`, and perhaps `transferFrom` (though ERC-1155 has `safeTransfer` built-in). Additional contracts may include a **Marketplace Contract** for point swaps, and an **Upgrade/Governance Contract** if we implement upgradability or configurations. We might also deploy a **“Loyalty Manager” contract** that owns the token contract and provides higher-level methods corresponding to business actions (e.g., a single function call to handle redemption which might involve transferring tokens from customer to merchant and emitting an event Toast's system listens to for confirmation). In some academic designs, a “loyalty manager” contract coordinates multiple mechanism contracts[37] – in our case it might not be necessary unless we separate logic for different loyalty mechanisms.
- **Customer Wallets:** Each loyalty program user will need a blockchain wallet address to hold their tokens. However, we cannot expect mainstream restaurant patrons to handle private keys and crypto wallets in the traditional sense. Therefore, Toast will implement a **custodial or semi-custodial wallet solution** linked to user accounts. Options include:
 - *Custodial:* Toast generates and holds the private keys in a secure vault (possibly partitioned by user or using an HSM). The advantage is a completely seamless experience – users login with their email/phone as they do now, and Toast manages all blockchain transactions under the hood. The downside is users don't directly control their tokens outside the platform (though we could allow exports).

- *Integrated Key Management:* We could use a service like Magic.Link or Web3Auth, where user keys are derived from their login credentials or stored in the cloud but accessible by users. Another approach is smart contract wallets (e.g. Argent, Gnosis Safe) where user login triggers meta-transactions.
- *Non-custodial (power users):* We may allow advanced users to link an external wallet (e.g., Coinbase Wallet or MetaMask) to their Toast account if they want to self-custody. But this will be an option, not mandatory.

Initially, we likely go **custodial for ease**. Each user gets an address (we maintain a mapping userID -> walletAddress in our database). The Loyalty Middleware knows the private key for that address (or accesses it when needed) to sign transactions. When we mint points, we might mint directly to the user's address. For redemptions, a user's signature may be needed to transfer their tokens to the merchant or burn; to avoid requiring users to sign with keys every time (which is impractical at a POS), we will implement **meta-transactions**: the user can pre-authorize Toast (via terms or an in-app approval) to perform loyalty contract actions on their behalf. Technically, we could have the smart contract allow an "operator" (Toast's address) to move tokens for users – ERC-1155 supports operator approvals. Upon account creation, we could set Toast as an operator for the user's tokens. Then, when a redemption happens, the Toast service can execute the transfer from the user's address without the user physically signing a transaction each time. This approach was seen in the research where a loyalty manager contract enabled customers to transact without fees by using meta-transactions and an operator model[37].

- **Merchant Wallets:** Each merchant (or each restaurant location, though probably aggregated at brand level) will also have a blockchain account. This is used to **receive tokens on redemption** if we design it that way, and possibly to hold any reserved tokens for promotions. For example, if customer uses 50 points to pay for part of their bill at Restaurant A, those 50 tokens could be transferred from the customer to Restaurant A's address (or to a sink address). If transferred to Restaurant A, it accumulates tokens that Restaurant A had originally issued – basically their own points coming back. The restaurant could choose to hold those (as proof of how many points have been redeemed, useful for accounting) or burn them immediately via the contract (reducing total supply). The merchant wallet would be controlled by the merchant but realistically most will let Toast's system handle it (so Toast might also custodially manage merchant keys, unless a merchant requests holding it). For security, merchants might have multi-sig control if needed (especially larger chains with finance departments wanting oversight on loyalty liabilities).
- **Toast Off-Chain Database:** We will continue to maintain an off-chain database that tracks user point balances, transaction history, etc., synchronized with the blockchain. This is critical for performance and reporting. For instance, a POS terminal querying "how many points does this customer have available?" should

get an immediate answer from Toast's database (which is updated in near-real-time from blockchain events). Relying on the blockchain directly for every query would be too slow and complex for the POS. Therefore, the Loyalty Middleware or a separate **Blockchain Indexer** service will listen to all token events and update the database. This also allows us to show consolidated info to users (perhaps converting everything to a common value for display). The database acts as a cache and system of record in parallel to the blockchain. In case of any discrepancy, the blockchain is the source of truth for balances, but our goal is to keep them in sync always.

- **Integration with Toast Systems:** The architecture will integrate with various existing components:
- The **Toast loyalty management interface** (used by merchants to configure loyalty program settings) will be extended to include blockchain-specific settings (like toggling on/off trading for their points, viewing on-chain stats, etc.). However, the fundamentals (setting point earn rate, reward catalog) remain similar.
- The **Guest-facing interfaces** (Toast TakeOut app, online portals, or receipts) will be updated to display token balances and possibly a QR or code representing the user's wallet for scanning. For example, a digital receipt could include "You earned 50 ToastToken[RestaurantXYZ-2025]. View on blockchain: [link]" to emphasize transparency.
- The **Settlement and finance systems** will get new reports showing liabilities per token type, redemptions, expirations, etc., drawing from both on-chain data and our accounting rules.

Process Flow Examples:

- *Earning (Issuance):* When a sale is completed, the POS sends an "earn points" request to Toast's loyalty API with the user ID and amount spent. The loyalty service calculates the points (say 100 points) per that merchant's rules. It then calls the blockchain middleware: `mint(userWallet, merchantTokenID, 100)`. The middleware signs a transaction calling the ERC-1155 contract's mint function. Within seconds, the transaction is mined on the blockchain, creating 100 new tokens in the user's wallet address. The middleware receives the event and updates the Toast DB: user's balance for that token +100, and records transaction details. The POS can immediately show "100 points earned" (optimistically, even if the blockchain tx is in process, we know it will happen; we can later reconcile any failures). Because the blockchain is the ledger, double-earning or tampering is prevented – the contract won't mint more than authorized, and every issuance is logged.

- *Redemption:* A customer wants to use 200 points to pay for part of their bill. On the POS, the cashier selects "Redeem 200 points". The POS contacts Toast loyalty service: `redeemPoints(userID, merchantID, 200)`. The service will verify the user has 200 available (in DB) and then call the blockchain: either a transfer from user to merchant or a burn from user. If using operator privileges, the Toast middleware (as operator for user) calls `burn(userWallet, merchantTokenID, 200)` on the ERC-

1155 contract[39]. Those tokens are destroyed on-chain. Alternatively, `safeTransferFrom(user -> merchant, 200)` could be used to move them to merchant. In either case, the contract emits an event for the redemption. The middleware confirms transaction success and then instructs the POS that the points are redeemed, so POS applies a \$2.00 discount on the bill (if 100 points = \$1). The receipt prints the discount and new point balance. All this needs to occur quickly – our middleware likely will execute the blockchain transaction *after* instantaneously guaranteeing the POS that the points are deducted. We can achieve real-time speed by using an off-chain confirmation: e.g., Toast marks those 200 points as “spent/pending” in the DB immediately and lets the sale continue, while the actual token burn may finalize a few seconds later on-chain. This asynchronous pattern is acceptable as long as we ensure eventual consistency (if the blockchain transaction were to fail for some reason, we’d have a reconciliation process, though that should be rare if our contract logic is correct and we have proper error handling).

- *Transfer/Trade*: For a user-to-user gift, the flow might be initiated in the app: User A selects friend User B and an amount to send. The app calls Toast service, which then either uses an operator transfer (moving tokens from A to B on-chain) or generates a unique link/QR that encodes a signed transfer for B to claim. The operator method is simpler: Toast’s service calls `safeTransferFrom(A -> B)` on the contract (Toast being approved to do so). The DB updates both A and B balances. For a marketplace trade, the flow involves the marketplace contract: both users (or Toast on behalf) would call the contract to lock their tokens and the contract swaps them when conditions meet (this is complex under the hood but abstracted to users as a simple “trade X for Y” action). Post-trade, events update balances.
- *Meta-transactions & Gas*: One goal is **gas abstraction** – we don’t want users to have to hold Ether/MATIC/etc. to pay transaction fees. We plan to either run on a no-gas network or have Toast cover the gas. Meta-transaction frameworks allow the user to sign a message saying “I want to do X”, and a relayer service (run by Toast) submits it to the blockchain with its own funds, thus the user pays no fee. Our loyalty contract can integrate with OpenZeppelin’s GSN or a custom meta-tx relay to support this[37]. If using a permissioned chain, we can eliminate gas or use an internal gas token that Toast pre-distributes. In any case, the user experience will be that performing loyalty actions does not require them to know about or spend cryptocurrency.

Security considerations in the architecture include using secure communication between POS and cloud (as currently), securing the private keys in the middleware (likely in an HSM or using a key management service), and thorough authentication checks on any API that triggers blockchain actions (only the authorized Toast systems can mint/burn, and only according to business rules). Smart contracts will be audited to ensure no vulnerabilities (like someone could mint tokens without authorization, etc.). We will also implement

monitoring – the distributed ledger can be monitored for unusual activity (e.g., large transfers, etc.) in real-time.

In summary, the architecture extends Toast's platform with a parallel blockchain ledger for loyalty points, while insulating end-users and merchants from blockchain complexity. It provides the benefits of decentralization (shared source of truth, security) without sacrificing the speed and usability of Toast's existing cloud system. The **figure above** shows the flow: the Toast POS communicates with loyalty middleware, which in turn interacts with smart contracts on the blockchain. Customers have wallets (largely managed by Toast) and merchants have accounts, with Toast's database tying it all together for query and reporting purposes.

This architecture is designed to be scalable as the program grows. We can start with a relatively centralized model (Toast running most pieces) and over time consider decentralizing some aspects (for instance, allowing multiple validators if we ever form a consortium with big restaurant partners, akin to each partner co-owning the network nodes[15]). The immediate next step after design approval would be developing a prototype of this architecture to validate performance (ensuring that even during busy lunch hour, minting points for transactions doesn't bottleneck and redemption is instant from the cashier's perspective).

6. User Experience Flow

The success of the blockchain loyalty program will depend heavily on a frictionless user experience. Our aim is to abstract the blockchain layer away so that from the customer and staff perspective, using loyalty points is as easy (or easier) than today. Below, we outline the end-to-end user journey and how each interaction will work in the new system:

- **Enrollment:** Customers can enroll in the loyalty program through existing channels (e.g. via a tablet at the restaurant, the receipt QR code, Toast mobile app, or an online signup). This process will remain simple: providing a phone number or email and agreeing to loyalty T&Cs. Behind the scenes, once enrolled, the system will **create a blockchain wallet** for the user (if one doesn't already exist). The user will not need to manage a key – the wallet address is linked to their Toast account. If the user already had an account, the wallet is created and associated automatically. We will make this invisible to the user; however, we might introduce the concept by giving them a “digital loyalty card” identifier which in reality corresponds to their wallet. For example, in the app it might show a loyalty membership number or a QR code that encodes their wallet address for quick lookup at POS. Importantly, the signup confirmation can mention that their rewards are stored securely and transparently on a blockchain (an optional educational point that could increase perceived trust: “Your Toast Points are now secured on a blockchain ledger for your peace of mind[13].”)
- **Earning Points (In-Store Purchase):** When making a purchase at a participating restaurant, the customer identifies their loyalty account (by scanning their app QR

code or providing their phone number to the cashier, as is common). The cashier finalizes the transaction. Immediately, the POS will show something like “You earned 50 points today!” and the customer might also get a push notification on their phone: “You earned 50 ‘ToastPoints’ from Joe’s Diner.” The experience is similar to before, except now those points are tokens. The user can open their Toast app and see the updated balance. For clarity, the app might list points by merchant (e.g., “Joe’s Diner: 150 points, Sushi Place: 80 points”). The app will likely abstract vintages unless the user digs into details; it would just ensure they use older (earlier vintage) points first when redeeming. If the user taps on a specific balance, we could display extra info like “120 points expiring 12/31/2025, 30 points expiring 12/31/2026”. We will emphasize new capabilities: an in-app tooltip or tutorial can explain “Your points are now tokens – you can even send them to a friend or use them across our network!”

- **Viewing and Managing Points:** The Toast user app/website will have a **dashboard** for loyalty. This will show total points, maybe a dollar-equivalent if applicable. It will also show **activity history**, and here’s a new feature: we can show on-chain verified status (e.g., a checkmark meaning the blockchain transaction is confirmed). For power users, we might offer a “View on Blockchain” link for each transaction which opens a block explorer showing that token transfer[13][40]. This transparency can be a selling point, but it should be unobtrusive for those who don’t care. Users will also have controls: “Transfer Points” and “Trade or Swap Points” within the app.
- **Transferring (Gifting) Points:** Suppose a customer wants to gift some points to a friend or family member. In the app’s loyalty section, they’ll click “Send Points.” They choose the program (merchant) or overall points they want to send and enter the recipient’s identifier (perhaps phone number, or if they are contacts and both have Toast accounts, we can list them). For example, Alice wants to send 100 “Coffee Cafe” points to Bob. She selects Bob from her phone contacts (the app can integrate contacts to find others who have Toast loyalty accounts, similar to how payment apps find friends). She hits send; a confirmation screen might say “You are sending 100 Coffee Cafe points (worth \$5) to Bob. This will reduce your Coffee Cafe point balance and increase Bob’s.” Alice confirms. Bob then receives a notification: “Alice sent you 100 Coffee Cafe points!” Bob can see those in his app immediately. If Bob didn’t have a Toast account or that loyalty program, we’d invite him to sign up via the message (the system could have generated a claim link). But since the question context is mostly within the Toast ecosystem, we’ll assume both are enrolled. This gifting feature adds a social and convenience factor—families can share points, and it might encourage network growth (people inviting others so they can send/receive points).
- **Marketplace Trading (Advanced Users):** In the app, there will be an option “Swap Points” or “Marketplace.” This takes the user to a section where they can exchange points between programs. We will design this with a **simple UI** to avoid overwhelming users with the concept of order books or exchange rates. One

approach is a **swap wizard**: user selects “I have [dropdown of their point balances] and I want [some other brand’s points].” The app will then show a quote, like a currency exchange: “Current rate: 2 Sushi Place points = 1 Steak House point.” If the user agrees, they click swap, and the trade executes. Alternatively, if they want cash (if we permit conversion to a gift card or maybe to the Toast universal token convertible to cash), they might see “Your 1000 points could be sold for approximately \$8.50 (after fees)”. We likely won’t offer direct cash-out at first to avoid money transmitter issues, but we may partner with a third party for that eventually (similar to how some airline miles can be sold). The marketplace could also highlight offers: e.g., “This week: Get 20% extra if converting your points to Dave’s BBQ points!” (if Dave’s BBQ runs a promo via the marketplace). The UX will emphasize *usefulness* of trading: “Got points you won’t use? Swap them for ones you will!” Perhaps we integrate some intelligent suggestions: if the system notices a user has points at a place they haven’t visited in a year, it might prompt them, “You have 300 points at Taco Spot. Trade them for another reward you’ll use.” Under the hood, trades might not be instant matches if low liquidity, so we might initially operate it more like a **universal exchange** where Toast itself intermediates at published exchange rates (derived from average redemption values and any promotions). Over time, as volume grows, we can let market dynamics set the rates through buy/sell offers, which advanced users might engage with. For most, it will feel like a simple currency converter inside the app.

- **Redemption (Customer Perspective):** When a user wants to redeem points at a restaurant, the process remains intuitive. At ordering time, the cashier might ask “Do you want to use your points?” or the user might inform them. The POS can show the available point balance (thanks to the integration). If the user says yes, either they specify how many to use or use maximum. After applying the points, the receipt will show the discount and possibly remaining points. From the app perspective, the user will see their balance drop accordingly, near real-time. They might also get an immediate alert: “You used 50 points at Joe’s Diner, saving \$5 on your bill. Your new balance is 150 points.” If the user has multiple types of points that could be used (say we later allow using generic points anywhere), the app or POS may allow choosing which points to apply. But initially, if each merchant’s points only redeem at that merchant, it’s straightforward. For online orders or self-service kiosks (if Toast supports those), the app or web checkout would have a “Apply points” button doing the same thing. We ensure that partial redemptions and split payments (points + cash) are handled, as is typical in loyalty systems. One advantage to highlight: because everything is tokenized, even if the user doesn’t have enough points for a full reward, they can use what they have (like spend 50 points for \$5 off rather than needing 100 points for a specific free item, depending on how merchants structure it). This aligns loyalty with a cash-like flexibility.
- **Multi-Merchant Use Case:** If in future we allow cross-merchant redemption (like one currency for all), the user experience would be: user goes to Restaurant B and

the app/ POS shows “You have X points from Restaurant A that can be used here via conversion.” Maybe they click convert and use. But if we do not allow direct cross-use at POS (which complicates settlements), users would need to do a swap beforehand in the app (which they could even do on the spot if they realize they have other points they want to use). We can streamline that by integrating the swap suggestion: e.g., at checkout, POS says “You have no B points, but you have A points worth \$Y, would you like to swap and use them?” If user agrees, behind scenes a trade happens, then redemption. This kind of fluid experience would be phase 3 or 4 presumably.

- **Feedback and Gamification:** The app can use the blockchain aspect as a marketing plus. For example, we could give users **badges (NFTs)** for certain achievements – these NFTs could be another token type in the ERC-1155 contract (non-fungible per user). E.g., “Blockchain Loyalty Pioneer – first 1000 users” badge, or merchant-specific badges for being a top customer. These could be shown in-app and even tradable if we wanted (some brands might create collectible rewards). This adds fun and also educates users that they truly own these digital items. We might also show a “market value” of their points portfolio (like “Total points across all brands: 5,000 (~\$50 value)”). Seeing a dollar value could increase user engagement, as it makes the rewards feel tangible, almost like a balance in a bank or crypto wallet.
- **Customer Support & Recovery:** One UX consideration is if a user loses access (e.g., they get a new phone and forgot their login). Since we custodially manage, they can reset password and we still have their points – much better than if it were purely user-custodied and they lost a key. If a user believes their account was hacked or points stolen, we have on-chain records to investigate. Potentially, if theft occurred (which would likely require their Toast account to be compromised), we might have the ability to freeze transfers of those tokens or reissue (since it’s a permissioned system we can theoretically mint replacement and blacklist certain addresses, though we’d use that power sparingly to maintain trust in immutability). We will have a process for handling fraud claims similar to fraud in gift card or loyalty today.
- **Merchant & Staff Experience:** (Though this section is “User”, staff are users in a sense too.) For cashiers and managers, the process should not introduce extra steps. The POS will automatically query and apply blockchain loyalty info via Toast’s normal interface. Training staff might simply involve telling them “Points now work a bit differently but you use them the same way on the screen.” They might notice wording changes (maybe the POS screen says “Blockchain ID” instead of “Account number” or something technical, but likely we hide that). We will ensure redemptions are as quick as credit card auth times or faster. If the internet or blockchain network is down at that moment, we will have fallback (perhaps allow an offline issuance logged later, or a cap on offline redemptions – similar to how some systems allow one offline transaction trusting it and later reconciling). These

edge cases will be addressed to avoid any embarrassing outages where “loyalty can’t be used” – the core dining experience must remain smooth.

In summary, the user experience is designed to **feel familiar**, with added capabilities: - Enroll, earn, and redeem are as easy as before (with messages highlighting transparency and security improvements). - New features like sending points to friends and swapping between rewards will be available in-app for those who want them, but won’t complicate the experience for those who don’t. - The app will serve as the hub, showing all loyalty assets in one place (solving the fragmentation issue where before a user might have had to keep track of each restaurant’s points separately – now one interface shows it all). This consolidates value for the user[41]. - Throughout, terminology will be carefully chosen: we might call them “Toast Points” publicly or keep each brand’s name. Possibly “Toast Points by [Merchant]” to emphasize the network while retaining individual brand identity. Marketing can decide a catchy name (“Toast Tokens” or similar, if that branding is acceptable). We’ll ensure users understand these are still loyalty points, just supercharged with new powers, not some volatile cryptocurrency they need to worry about.

With these user flows, we expect increased engagement: more sign-ups (due to word-of-mouth that points are more useful), more active use of points (since they can trade or save them without fear of waste), and a smoother omnichannel experience (online/offline unified via the blockchain ledger).

7. Rollout Phases

Implementing this initiative will be a significant project touching technology, operations, and stakeholder adoption. A phased rollout will help manage risk and incorporate feedback at each stage. We propose the following rollout phases:

Phase 0: Research & Planning (Q4 2025) – In this preparatory phase, we will finalize system design, select the blockchain infrastructure (evaluate Ethereum L2 vs private network), engage legal/compliance for necessary approvals, and possibly run focus groups with a few restaurants and customers to validate the concept. We will also develop a detailed project plan and form an internal task force (involving engineering, product, legal, support, and selected merchant reps). Additionally, we might create a small prototype to internally simulate earning/redeeming tokens to ensure our approach works as expected. Success criteria for this phase: executive sign-off on design, legal green-light or clear path identified, and merchant champions identified for pilot.

Phase 1: Technical Pilot (Q1 2026) – Build the core blockchain loyalty platform and test it in a controlled environment. This phase involves: - Developing the **smart contracts** (ERC-1155 token contract and basic marketplace contract) and getting them audited. - Implementing the wallet system and integrating it into a test version of the Toast app. - Upgrading a test POS environment to use the new loyalty API calls. - Selecting one merchant (possibly a single restaurant or a small local chain that is friendly to trying new features) to participate. Alternatively, Toast could pilot it in an employee cafeteria or a pop-

up cafe to simulate usage. - Running the pilot with a limited set of “users” (could start with internal employees or friends/family as the loyalty customers to shake out issues). For example, Toast could create a mock restaurant on the system where employees get points and trade them amongst themselves as a test. - During this time, monitor performance: ensure that minting and transferring on the chosen blockchain is working within acceptable time. We’ll test failure modes (node goes down, etc.) and security. - No public customers yet, to allow iteration. The objective is to validate end-to-end functionality: from POS transaction -> blockchain -> app updates -> redemption. We will collect feedback from pilot users and the merchant staff: Was the experience smooth? Did the app make sense? Any confusion or errors? Technical metrics like transaction throughput and any blockchain fees will be evaluated to tune the system.

Phase 2: Limited Beta Launch (Q2–Q3 2026) – Expand the pilot to a real-world setting with actual customers, but still limited in scope. We will: - Recruit a small group of **merchant partners** (perhaps 5-10 locations) to introduce the blockchain loyalty program to their customers. Ideally, pick a range: for example, a cafe, a quick-serve restaurant, and a mid-range restaurant in different regions. Possibly merchants who are already using Toast loyalty actively and have expressed interest in innovation. - Provide training and support to these merchants. Roll out an update to their Toast POS that has the integrated loyalty blockchain functionality. - Invite their loyalty customers to participate. This might be done via an email explaining “We’ve upgraded our loyalty program – your points are now even better!” highlighting new features. For transparency and excitement, the communication can mention the points are on “a secure blockchain, allowing you to exchange points with other customers or even use them across participating restaurants.” Some early adopter customers will find that cool; others might not notice. - Monitor usage closely. This is a beta, so we might impose some limits (maybe disable external trading initially, focusing on in-restaurant earn/redeem and maybe simple gifting between beta users). We’ll have our development and support teams on standby for any issues. - Solicit feedback regularly from both customers and merchants. Perhaps set up a forum or regular call with the merchant managers to hear their thoughts. Also track any support tickets or confusion points among customers.

During Phase 2, we also focus on **metrics**: enrollment rate (do more people sign up when they hear about the new program?), redemption rate changes, any changes in spending behavior (do customers spend more knowing points are more flexible?), etc. We also verify that our accounting processes (liability tracking etc.) function correctly in a live scenario.

If any adjustments are needed (e.g. UI changes, contract tweaks, performance scaling) we do them in this phase.

Phase 3: Feature Expansion + Official Launch (Q4 2026) – Based on beta results, finalize the product for broader release: - Add any deferred features such as the full **Marketplace trading platform** if it wasn’t active in beta. Ensure KYC/AML processes (if needed) are in place for that. Possibly integrate with a third-party exchange or loyalty network if beneficial. - Harden the system for scale: deploy more robust infrastructure (multiple

blockchain nodes for redundancy, maybe move from a pilot L2 to a more scalable network if needed), and do load testing for expected transaction volumes (peak times when many transactions might mint points simultaneously). - Conduct a security review and compliance audit before launch. This might include an external audit of the whole system (smart contracts, cloud integration, etc.), and obtaining any needed regulatory sign-offs (for example, if considering states that require registration for gift card-like programs, ensure we've done so). - Prepare comprehensive documentation and training for merchants and customer support teams. The documentation will include how the loyalty program works in the new model, FAQs (e.g., "What does it mean that my points are on a blockchain?"), and troubleshooting guides. - Launch marketing campaigns for the new program. At this official launch, Toast can do PR — highlighting that Toast is launching a **"Tradeable Loyalty Points Program on Blockchain"**, which could garner media attention in both the restaurant industry and tech press. That buzz can attract new restaurants to Toast and excite customers of participating merchants. - Roll out the updated Toast app to all users with the loyalty section features enabled. Also push the POS updates to all Toast merchants (or those opting in initially). - **Opt-in vs Opt-out:** We must decide if all Toast loyalty clients are automatically switched to the new system or given a choice. To mitigate risk, we might do an opt-in program at first: merchants can choose to upgrade to "Toast Loyalty Plus (Blockchain)" or stick with legacy for a while. However, running two systems in parallel long-term is inefficient. Likely we will migrate everyone eventually, but doing so in waves (maybe by region or merchant size) could be Phase 3 and Phase 4 steps.

Phase 4: Network Growth (2027 and beyond) – After the initial launch, the focus shifts to growing adoption and enhancing the ecosystem: - Enroll more merchants: Use the success stories from Phase 2-3 beta to onboard the rest of Toast's merchant base. Our sales and account managers can pitch the benefits ("higher customer engagement, part of a network, marketing opportunities") to get restaurants excited to join. Possibly provide incentives such as a period of discounted fees or bonus points funded by Toast for early adopters. - Foster cross-merchant promotions: Now that many are on the network, we can facilitate special campaigns. For instance, "Dine at any 3 participating Toast loyalty locations in a month, get a bonus 100 tokens." The blockchain system makes multi-brand tracking easier, as all points can be seen collectively. We might also introduce a **coalition reward** (like a free item redeemable at any of a set of merchants, essentially an NFT coupon) as a next feature. - Continuous Improvement: Gather ongoing feedback, and improve features like the marketplace UI, add dynamic pricing or auctions for points, maybe integrate a **tier status** (some programs use NFT-like tokens to represent tier status that are tradable – though that has its own implications). - Global/Enterprise Expansion: Evaluate launching the loyalty token program internationally if Toast operates in other countries, adjusting for local laws. Also, consider partnering with large enterprise clients: for example, if a restaurant chain on Toast wants its own branded token but still interoperable, we can accommodate that in our system (just with a unique token ID and maybe co-branded name). - Monitoring and governance: As the ecosystem grows, possibly form a governance committee with some merchants to discuss any proposed changes to the program rules or smart contract upgrades – this is part of Phase 4 operations.

Throughout all phases, **risk management** is key: we will have a rollback plan (for instance, if something goes wrong, we can temporarily revert to issuing centralized points until fixed), and an open communication line to merchants and users to maintain trust.

A timeline summary: - 2025 Q4: Finalize design, legal prep. - 2026 H1: Build & pilot internally. - 2026 H2: Beta with select merchants and users. - Late 2026: Soft launch to more merchants, marketing push. - 2027: Full roll-out, with continuous feature enhancements and onboarding.

This phased approach ensures we walk before we run – proving out the concept in small scale, gradually scaling up once we’re confident in the technology and user reception.

8. Tenant Economics

A crucial aspect of this proposal is how it impacts the economics for participating restaurants (“tenants”). The goal is to ensure that the blockchain loyalty program is financially attractive to merchants, aligning with their business objectives of driving repeat visits and increasing customer spend, without introducing undue cost or liability. Below we analyze the economic implications and design considerations for merchants:

- **Cost of Loyalty Rewards:** In the current model, when a merchant issues loyalty points, it’s effectively giving a future discount (a cost to them when redeemed). That fundamental cost remains – blockchain doesn’t change that a point redeemed is a free item or discount the merchant provides. However, by enabling higher redemption (points are tradeable so more likely to be used rather than wasted), merchants could see an **increase in redemption rate**. For instance, if previously 20% of points expired unused (breakage), now breakage might drop significantly as unused points can be sold or given to someone who will use them. This means merchants might end up honoring more rewards. That sounds like a higher cost, but it also means the loyalty program is doing its job – those points now bring a customer in to redeem (who likely spends beyond the value of the points). Ideally, increased redemptions correlate with increased visits and revenue. We will need to help merchants model this: e.g., if breakage drops by 10%, how much more in discounts will you give versus how much extra revenue from those additional visits? Early studies in loyalty-blockchain suggest the improved engagement offsets the increased redemption liability because of higher customer retention[\[42\]](#)[\[43\]](#). To be prudent, merchants might choose to adjust earn rates slightly if needed to balance economics (for example, if points become more “valuable”, they might issue slightly fewer per dollar spent to keep effective reward percentage the same). We’ll provide them tools and recommendations for such tuning.
- **Cross-Merchant Value & Customer Acquisition:** One exciting aspect is the potential for **customer sharing among merchants**. In a coalition-like environment, a customer who frequents Restaurant A could be enticed to try Restaurant B because they have points they can use there (via trading or a universal acceptance model). From Restaurant B’s perspective, they might effectively “acquire” a new

customer at the cost of accepting some points as payment – analogous to accepting a gift card or coupon funded by someone else. This can be positive if it fills seats and leads to new regulars. Tenant economics in such cross-use scenarios usually involve compensation: Restaurant B would want to be reimbursed for the value of points from Restaurant A that were used at B. Our design currently uses trading rather than direct cross-redemption, meaning by the time points are used at B, they have been converted into B's own points (through a trade where another user or the market provided B points). Thus, Restaurant B is actually redeeming its own points (which they will account for as normal). The user who originally had A's points got rid of them in exchange, so Restaurant A's liability is transferred to whomever took them. If that was another user, eventually someone will redeem them at A or trade further. **Economic Outcome:** Each merchant ultimately only bears the cost of their own points being redeemed at their location, just like today. Trading doesn't create new value out of thin air; it just reallocates who holds the right to claim that value.

- **Market Price of Points:** One question merchants might have is “Will my points trade at a discount? What does that mean for me?” In an open market, if one merchant's points are less in demand, users might sell them at a lower price (e.g., \$0.005 per point even if face value is \$0.01). This could happen for businesses with lower perceived value or if people earn a lot of points they don't want. While it might be uncomfortable for a merchant to see their points “discounted” publicly, it's not necessarily bad – it means deal-seeking customers can buy those points cheaply, effectively getting a discount to try that restaurant, which is a marketing opportunity. In fact, loyalty points already have an implicit market value – e.g., on forums people might value airline miles at X cents each. We're just making it explicit. We should reassure merchants that a lower trading price doesn't equate to lost revenue directly; it's more akin to offering a promotion to attract customers. The merchant still controls the face value (how points translate to rewards). If the market price is too low and they dislike that, they could respond by enriching their program (making rewards more generous, which increases demand for points), or possibly buying back points as mentioned earlier to reduce supply. Conversely, if a merchant's points trade at or above face value, it indicates strong demand – perhaps their capacity might become an issue if everyone rushes to redeem (a “good problem” indicating popularity). Toast can provide analytics to merchants: “Your points are trading at 80% of face value, meaning some customers value them slightly less – consider boosting reward offerings or partnering with other tenants to increase desirability” or “Your points are in high demand on the exchange, trading near face value or higher (if there are exclusive experiences driving up value).”
- **Incentives and Funding:** Initially, to kickstart the ecosystem, Toast might bear some cost or provide incentives. For example, Toast could fund a **welcome bonus**: give each user some free points or a small amount of a universal token when they sign up, to encourage usage. Or Toast could subsidize some trades (cover

transaction fees or provide a match: trade 100 points and Toast gifts an extra 10, for instance) to stimulate liquidity early on. These are essentially marketing costs that Toast or participating merchants might share. Part of tenant economics is deciding if merchants contribute to a joint promotion pool. We could set aside a portion of our transaction fee revenue from the marketplace into a “loyalty growth fund” that periodically rewards active customers or high-performing merchants in the network.

- **Merchant Fees:** Currently, Toast likely charges a fee for the loyalty software module (either a flat subscription or usage fee). With the enhanced program, Toast could justify a **premium loyalty fee** or marketplace commission. We must ensure any fees are outweighed by value. For instance, if Toast were to take a 1% cut on point trades, this should be reasonable given we provide the platform – likely this fee would be paid by the trader (user) rather than the merchant. We might not charge merchants more initially to encourage adoption, but in future, if this drives significant incremental sales for them, Toast could consider a small revenue share or higher SaaS fee for the advanced loyalty features. It will be important to clearly communicate ROI: e.g., if a merchant pays \$X/month for this feature but sees Y% increase in customer return rate, that should more than pay for itself. Data from pilots will help make that case.
- **Liability Management:** Every point issued is a liability (deferred obligation to provide product). On the blockchain, we’ll have real-time visibility of how many points are outstanding for each merchant. This can actually help merchants manage that liability more actively. For example, if a merchant has an excessively high outstanding balance of points (maybe from a big promotion), they might take steps to encourage redemption (to clear it) or account for it financially. If points are expiring in 3 months and lots remain, they might run a campaign “Double value if you redeem now” to avoid a rush at the last moment or disappointment. These decisions remain similar to traditional loyalty but are aided by better data transparency.
- **Inter-tenant Settlement (if needed):** Our design avoids direct cross-redemption, but in case we enable something like “Spend your Toast universal points at any Toast restaurant,” then we’d implement a settlement system. Typically, in a coalition, merchants buy points from a central issuer or settle after redemption. For example, each point might be priced at \$0.008 to buy for issuance (so merchant effectively funds slightly less than face, expecting breakage or overhead covers the difference) and redeeming merchant gets \$0.01 of value or similar. Toast could act as the clearinghouse taking a spread. However, to keep initial implementation simple, we won’t do universal points until phase 2 or 3, if at all. The trading mechanism sort of decentralizes the settlement – the market decides the rate and effectively “settles” value between users. If and when we consider direct coalition points, we would formalize how merchants reimburse each other. Possibly via a formula or using the marketplace as an intermediary (merchant A’s points when

used at B are immediately sold to a pool which merchant A later pays into). For now, merchants should consider that **points remain their responsibility until someone else redeems them**. Trading doesn't erase their liability; it just moves who holds the claim. If a merchant's points are widely traded but ultimately no one redeems them and they expire, that merchant got the benefit of not redeeming (like breakage currently). If lots of people buy their points and redeem, the merchant sees more traffic and uses up the liability (as intended).

- **Upside for Merchants:**
- **Customer Acquisition:** As mentioned, being part of a larger network can bring in new customers. It's akin to joining a mall's gift card system where a mall gift card can be spent at any store – a store might see redemptions from customers who got a gift card intending to spend at some store but wander into theirs. The difference is we have an exchange rather than a common gift card, but the effect can be similar if customers actively trade to try new places.
- **Increased Spend:** Studies often show that when customers redeem points or gift cards, they tend to spend above the reward amount (e.g., a customer uses a \$5 reward but spends an extra \$20 on that visit)[\[44\]](#)[\[45\]](#). More redemptions can therefore boost top-line sales.
- **Cash flow from point sales:** If merchants choose to, they could **sell points for cash** either directly to customers (like buying points packages—though not common in restaurants, but maybe one day) or to other businesses in partnerships. For example, a hotel chain might buy restaurant points to give as rewards to their guests (“stay with us, get a free coffee at X”). Blockchain tokens make such B2B exchanges easier since there's a clear ownership transfer. This could be a future avenue: Toast could broker deals where one brand's points are used as incentives by another (with proper compensation). That essentially monetizes loyalty assets for merchants in ways not previously possible. JPMorgan and Affinity Capital's venture suggests turning loyalty points into an asset class for financing[\[7\]](#) – while that's advanced, our system lays groundwork for such innovation if merchants desire (they could potentially borrow against their token reserves or sell off liabilities to interested investors at a discount; those scenarios are outside current scope but become conceivable with tokenization).
- **Merchant Perspective & Adoption:** We recognize some merchants might be wary: the mention of “blockchain” and “tradeable” could trigger concerns about volatility, complexity, or loss of control. Our approach should reassure them:
- They **do not lose control** of their loyalty program – they still set how points are earned and what rewards they can get (those rules are enforced via smart contract and off-chain logic that only affects their token).
- They aren't giving away free stuff to other merchants' customers without reciprocation; trading is between customers.

- Compliance burdens (like tax on rewards or financial reporting) remain largely the same as before, just with better tooling.
- We should also highlight possible **marketing benefits**: Being part of the launch could give them PR exposure (“Featured as an innovative partner in Toast’s loyalty program”). Also, tech-savvy younger customers might gravitate to these restaurants because of the novelty and flexibility of the program (e.g., crypto enthusiasts who like the idea of tokens will support these businesses).
- **Small Merchant vs Large Chain**: Larger chains might have more ability to analyze and adapt to such a system (and may even have their own loyalty in parallel to Toast’s). Small independent restaurants might just follow Toast’s default settings. We need to cater to both with a program that works out-of-the-box but allows customization. For example, a chain could opt to let their points trade, while a small merchant can opt-out of marketplace trading if they feel it’s not for them (though that breaks some network effect, so we prefer they all participate, but offering choice might ease adoption anxiety initially). In time, once success is evident, all will likely opt-in as the standard.
- **Token Value Stability**: Since each point’s value in terms of goods is fixed by the merchant’s redemption policy, we don’t expect wild price swings like cryptocurrencies. However, to protect consumers and merchants from any manipulation or speculation extremes, Toast could implement safeguards: e.g., if a point’s market price moves beyond a certain band quickly, we could temporarily halt trades for that token (to investigate if something odd is happening). This is part of governance. But economically, as long as merchants honor a stable exchange rate of points-to-discounts (which they will), there is an intrinsic anchor to value.
- **Fraud Reduction**: The blockchain system might reduce certain fraudulent activities that cost merchants money – for instance, fake loyalty accounts or unauthorized point arbitrage. Every token is accounted for on-chain, and double-spending is impossible. Smart contracts can eliminate issues like someone trying to use the same points twice across systems (which might happen if data was out of sync in legacy systems). Less fraud is an economic win for merchants.

In conclusion, the design is intended to be a **win-win for merchants**: - They maintain the benefits of their loyalty program (repeat business, customer data) but now with enhanced reach and appeal. - Potentially attract new customers from the shared ecosystem. - Gain insights from marketplace data (e.g., if many people are selling their points, maybe their program needs improvement; if people are buying their points, that’s a positive indicator). - The costs (redemptions) increase in proportion to increased engagement – which is the point of loyalty marketing spend anyway.

We will articulate these economics clearly to tenants in an onboarding guide. By running pilots, we aim to gather concrete data: e.g., “In pilot, Restaurant X saw a 15% increase in loyalty redemptions, but also a 12% increase in overall spend from loyalty members, leading to net positive revenue impact.” Those figures will help convince others.

Finally, governance of economics: If merchants collectively feel something isn't fair (say one merchant is flooding the network with points that others feel devalue the currency), we might have a governance process to address it (discussed next). Transparency is the best remedy: every merchant can see on-chain who is issuing how many tokens, which fosters a sense of accountability and trust in the loyalty alliance[6].

9. Risk and Governance Checklist

Implementing a blockchain-based loyalty system introduces new risks and governance challenges that must be identified and managed. Below is a checklist of key risk areas along with mitigation strategies, as well as an outline of how governance will be handled to ensure the system's integrity and adaptability:

- **✓ Regulatory Risk:** *Risk:* Changes in regulation or misclassification of the token could lead to legal issues (e.g., being deemed a security, or subject to money transmitter rules). *Mitigation:* Consult legal counsel in all relevant jurisdictions during design. Clearly define token utility (no profits, only redeemable for goods) in user agreements[1]. Proactively seek no-action letters or regulatory sandbox programs if available. Remain flexible to geo-restrict or modify features (like disabling trading in regions that disallow it). Monitor legislative developments (such as any new laws on crypto assets or loyalty programs). Maintain an ongoing compliance committee.
- **✓ Security Risk (Cybersecurity):** *Risk:* The introduction of crypto wallets and smart contracts opens new attack surfaces. Hacks or exploits could lead to theft of points or unauthorized minting. *Mitigation:* Employ rigorous security best practices: smart contracts to be audited by reputable third-party firms, use established libraries (OpenZeppelin)[37], implement bug bounties to encourage ethical disclosure. Protect private keys using HSMs or managed key vaults; rotate keys if needed. Enforce least privilege – for example, the mint function might only be callable by a designated contract account that Toast controls, minimizing who can trigger it. Additionally, implement anomaly detection – if an abnormal mint or transfer occurs (outside expected patterns), automatically freeze that token type and alert admins before more damage.
- **✓ Operational Risk (System Performance and Uptime):** *Risk:* The added complexity of a blockchain layer could cause slowdowns or outages affecting the POS or loyalty operations. If the blockchain network is down or congested, transactions might not go through, impacting customer experience at checkout. *Mitigation:* Choose a reliable, scalable blockchain infrastructure with known performance (if using a public network, possibly one with high TPS or an L2 with guaranteed throughput). Deploy redundant nodes and use fallback mechanisms: if the blockchain call fails, the system should queue the action and not block the POS – e.g., allow the transaction and sync later (with measures to prevent double spend of points). Pilot under heavy loads to tune the system. Also, maintain the ability to

revert to “offline mode” where loyalty transactions are logged off-chain and reconciled when back online (this fallback ensures business continuity in worst-case scenario). Provide 24/7 monitoring of the blockchain network status and integrate alerts.

- **✓ Customer Risk (Adoption and Misuse):** *Risk:* Some customers might be confused or intimidated by the concept of blockchain or make mistakes (like accidentally sending points to the wrong person’s wallet if external transfers allowed). There’s also a risk of scams (someone tricking users into sending them points, etc.). *Mitigation:* Emphasize simplicity in UX (users don’t need blockchain knowledge to participate). Provide clear in-app confirmations (“Are you sure you want to send these points? This action is irreversible.”). If possible, allow a short undo window for accidental sends (e.g., we could program a time-lock on transfers that gives user 30 seconds to cancel – but that complicates immediate use, so maybe not default, but something to consider for non-immediate transfers). Educate users: include a Help Center section on the new loyalty program, explaining security tips (just like banks educate about phishing, we educate “Never share your account login, only trade through official app”, etc.). Build fraud detection to flag suspicious activities (like a user rapidly transferring points to many others might be a hacked account – we could pause and verify). Lastly, ensure our customer support is well-trained to handle questions about the new system and assist if issues occur.
- **✓ Merchant Risk (Acceptance and Fair Use):** *Risk:* Merchants may fear losing control or facing unintended consequences, like a competitor somehow exploiting the system or dissatisfaction if something goes wrong (e.g., pricing errors in trades, etc.). Also, if one merchant engages in misconduct (say intentionally creating tons of points outside of actual sales – though minting is controlled, but imagine collusion to game the system). *Mitigation:* Governance framework: Toast will likely act as the central authority at first – setting rules and being able to intervene (like freezing a merchant’s token if fraudulent activity is detected). We will have **Terms of Service for merchants** that using the loyalty network means agreeing not to abuse it (e.g., not issuing points not tied to real transactions unless via allowed promotions). If a merchant leaves the platform, we have a plan for their token (maybe allow 3 months for redemptions then retire). We can implement admin functions to disable minting for a merchant if needed (like if they go bankrupt or something). To get buy-in, involve merchants in a **Loyalty Advisory Council** – a governance group that meets periodically to discuss program direction. Initially, Toast will lead decisions, but we can give large stakeholders a voice so they feel ownership. Over time, perhaps move to a more decentralized governance (maybe not fully like a DAO, but maybe multi-signature control of certain contract upgrades with merchant reps involved).

- **✓ Financial Risk (Volatility and Liabilities):** *Risk:* If points become too “liquid,” might speculators or brokers come in and cause volatility? Could that undermine the program (e.g., points hoarded or dumped)? Also, if liability grows faster than anticipated, could it pose financial strain on merchants or Toast? *Mitigation:* We will likely restrict access to only genuine customers in early phases – it’s not like anyone can just show up and trade unless they have an account and points earned or legitimately purchased. This isn’t a cryptocurrency open to global speculators without participating in the ecosystem. So speculation is naturally limited by the closed loop nature (to get points, one must either earn by purchase, receive from someone, or possibly buy from someone who earned – but if someone external wanted to speculate, they’d have to buy points from users; they might only do that if they think they can profit, which is unlikely beyond small arbitrage, since points have a fixed redemption value cap). Regardless, to maintain stability, we could impose rate limits on trades (no single user can trade more than X points per day), and monitor marketplace pricing. If any point currency shows extreme behavior, Toast can intervene (pause trading for that token and investigate cause – perhaps a mistake in that merchant’s policy). On liability, we’ll frequently run reports and share with merchants. If liability is building up (meaning lots of points issued, not enough redemptions), we might encourage promotions to draw it down or consider adjusting earn rates. This is similar to current loyalty liability management but with real-time insights.
- **✓ Technology Evolution Risk:** *Risk:* Blockchain tech changes quickly. There might be future upgrades (Ethereum protocol changes, new token standards, etc.) or even obsolescence risk (what if the chosen chain goes out of favor in 5 years?). *Mitigation:* Build with flexibility: use a widely adopted standard (ERC-1155) and keep the possibility open to migrate if needed. For instance, if we start on a sidechain and later want to move to another, we could create a migration tool (snapshot balances, reissue on new chain, and retire old). Having an upgradeable contract helps to adapt to minor changes. We’ll follow blockchain industry developments closely and maintain a relationship with the platform we choose (like being in their enterprise program if available, ensuring support). Part of governance is also deciding when to upgrade or integrate new features (like if in future we want to integrate directly with some crypto wallets, etc., how to do so safely).
- **✓ Reputation Risk:** *Risk:* If anything goes wrong (security breach, regulatory action, major customer complaints), it could harm Toast’s reputation and that of participating merchants. *Mitigation:* Take a cautious approach, as we laid out in phases. Don’t oversell the feature beyond its proven reliability. Ensure clear communication – underpromise and overdeliver. In case of an incident, have a crisis communication plan: e.g., if there were a bug that caused some points to be lost, Toast might reimburse affected users in cash or credit to preserve trust, while swiftly fixing the bug. Transparency will be our friend: ironically, a blockchain system can boost reputation if run well, because it shows Toast is modern and

open. But we must be ready to openly address any issues, showing that the system's design isolates problems (for example, a bug in one merchant's token can be contained, not wreck the whole system).

- **Governance Structure:** Initially, **Toast Inc.** will govern the loyalty platform centrally – setting the rules (point valuation standards, expiration policies, fees, etc.), managing the smart contract upgrades, and operating the core infrastructure. We will form an internal governance board with representatives from relevant departments (product, engineering, legal, finance, support). They will oversee day-to-day operations and emergency handling (e.g., if an urgent contract pause is needed, they authorize it). For external governance, we propose establishing a **Merchant Advisory Council** consisting of a few trusted restaurant partners (especially ones from the pilot phase or key accounts) who can provide input on policies and changes. This council might meet quarterly to review program performance and suggest improvements. While not binding, their input will guide us to ensure the system meets merchant needs.

In the future, as the network matures, we might consider more decentralized governance mechanisms for transparency and fairness. For example, giving merchants a voting right on certain changes (weighted perhaps by their size or stake in the program). One could even imagine a scenario where the loyalty token holders (merchants and maybe customers via some representation) vote on certain proposals (like changing expiration periods or adjusting marketplace fee rates). However, that level of decentralization would only be pursued if it adds value; we won't complicate governance prematurely.

- **Smart Contract Governance:** We will use upgradable contracts (via proxy or controlled by a multi-sig) so that we can fix bugs or add features. The multi-sig could initially be controlled 2-of-3 by three senior Toast execs, for instance, to ensure no single person can maliciously upgrade. Eventually, that multi-sig could include a merchant rep or an independent trustee for added confidence. Any upgrade will be communicated to all stakeholders in advance (except emergency security patches, where we may patch immediately but still inform merchants after the fact with explanation). We'll document how often and under what circumstances we will consider upgrades (to avoid the perception that rules can change arbitrarily – stability is important for trust).
- **Checklist of Technical Governance Items:**
 - Manage the list of authorized minters (i.e., adding new merchants to the contract's internal registry).
 - Manage global parameters like point-to-dollar default rates or expirations.
 - Monitor marketplace fairness (no abuse by any single user or bot – we might need to ensure one user isn't cornering a market; if so, maybe impose more limits).
 - Ensure compliance checks (perhaps an annual compliance audit on the system to satisfy any external regulatory requirements).

- **Data governance:** keep user data secure and ensure on-chain data doesn't violate privacy (this we covered – e.g., using userIDs, not names, on chain).
- **Dispute Resolution:** If disputes arise (say a customer said they sent points but the other claims they didn't receive), we have the blockchain record to verify. But we need a formal resolution process – likely our customer support can resolve by looking at logs. For merchant disputes (e.g., Merchant A might claim Merchant B is somehow free-riding or a user exploited something), Toast will mediate using data. We'll incorporate dispute clauses in merchant agreements.
- **Risk of Low Adoption:** Lastly, a risk is that users or merchants simply don't use the trading features, resulting in minimal benefit. We consider that in risk because if only a few use it, the marketplace could be illiquid and disappointing. Mitigation: make sure to drive adoption via incentives and education. If initial uptake is slow, consider campaigns to spark it (for example, Toast could periodically buy some points from users to inject liquidity or encourage trades by offering bonus, etc.). Essentially jump-start the network effects so it reaches critical mass.

By systematically addressing each of these points in our project plan, we aim to launch the blockchain loyalty program with robust safeguards. This checklist will be revisited regularly throughout implementation. Each item will have an owner monitoring it (e.g., compliance team for regulatory, DevSecOps for security, etc.).

In conclusion, while there are non-trivial risks in pioneering this new approach, our careful planning and governance structure should mitigate them to an acceptable level. The result will be a well-governed, resilient loyalty ecosystem that can adapt and thrive, delivering value to Toast, our merchants, and our customers.

[1] [35] [36] [43] celebrusadvisory.com

<https://celebrusadvisory.com/wp-content/uploads/2020/01/Loyalty-Exchange-2019-0515-v1.7.pdf>

[2] [3] [8] [9] [10] [13] [14] [16] [17] [40] [41] Loyalty Program with Blockchain Technology in 2024 | Ultimate Guide

<https://www.rapidinnovation.io/post/blockchain-in-loyalty-programs-revolutionizing-customer-engagement>

[4] [5] [6] [15] [44] [45] Grow and Expand Loyalty Reward Programs With Amazon Managed Blockchain (Part One) | AWS for Industries

<https://aws.amazon.com/blogs/industries/grow-and-expand-loyalty-reward-programs-with-amazon-managed-blockchain-part-one/>

[7] [18] [19] [20] Blockchain Loyalty Programs - Sizzling Or Fizzling?: By Ketharaman Swaminathan

<https://www.finextra.com/blogposting/22380/blockchain-loyalty-programs---sizzling-or-fizzling>

[11] [12] A blockchain based solution to improve loyalty program with NFT in agribusiness

<https://www.oaepublish.com/articles/jsegc.2023.09>

[21] [22] [23] [24] [34] Customer Loyalty Programmes (IFRS 15) - IFRS Community

<https://ifrscommunity.com/knowledge-base/customer-loyalty-programmes/>

[25] [26] [27] [28] [29] [30] [31] ERC-1155 | An Introduction to Multi Token Standard Development

<https://blockchain.oodles.io/blog/erc-1155/>

[32] [33] [37] [39] Blockchain-Based Loyalty Management System

<https://www.mdpi.com/1999-5903/15/5/161>

[38] API change log - - Release notes - Toast platform docs -

<https://doc.toasttab.com/doc/relnotes/devPortalApiChangeLog.html>

[42] The reward point flow for the four main roles inside the platform. | Download Scientific Diagram

https://www.researchgate.net/figure/The-reward-point-flow-for-the-four-main-roles-inside-the-platform_fig2_338093486