FUTUREX

WHITEPAPER

Understanding Cryptographic Architecture Across the NPCI Ecosystem

An architect's guide to program-specific cryptographic infrastructure for India's payment, card, clearing, ATM, and Aadhaar-enabled services.

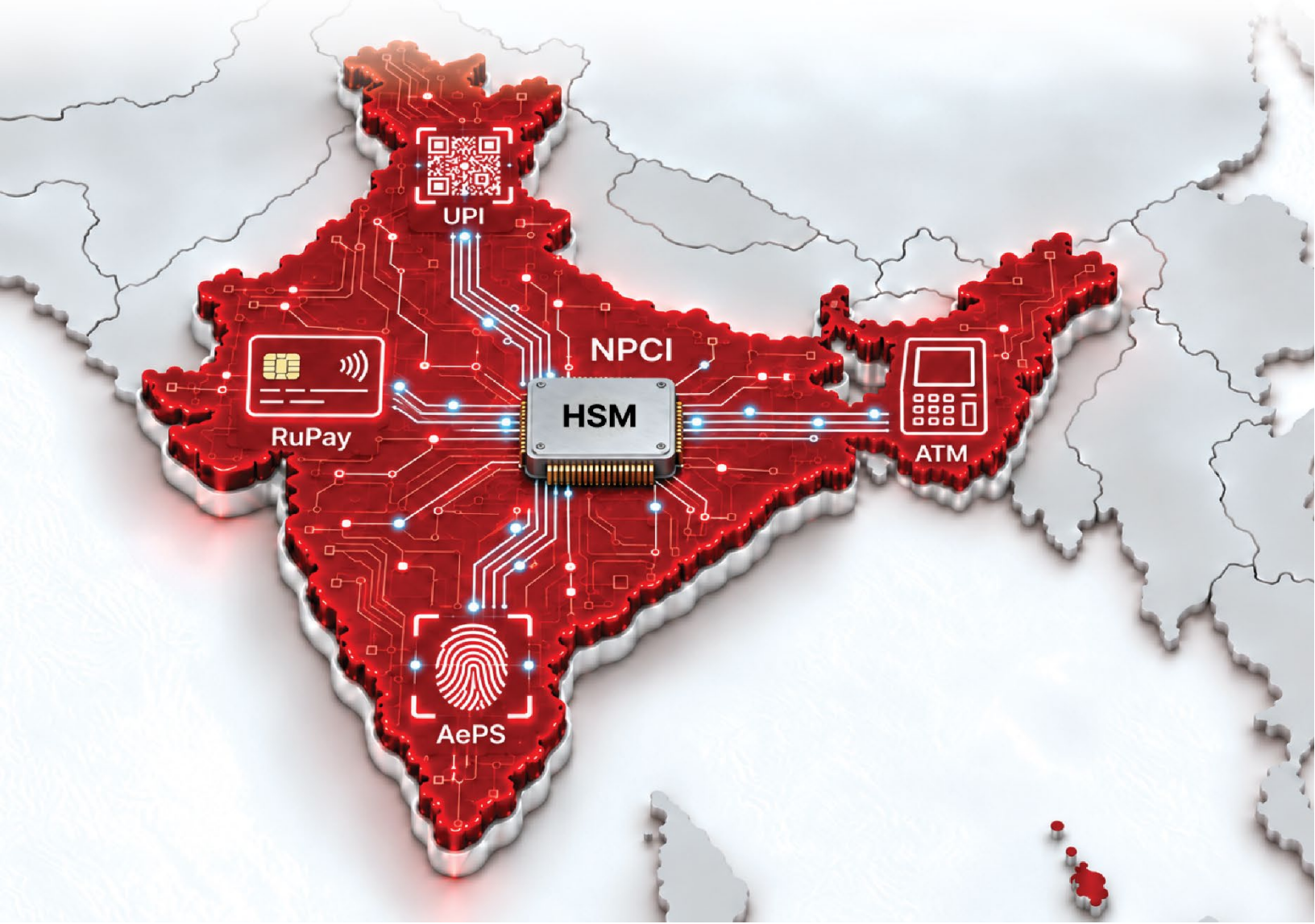


Table of Contents

Executive Summary 3
One Ecosystem, Ten Distinct Cryptographic Environments 8
One Organization Can Play Several NPCI Roles 10
Ten Program-specific Architecture Profiles 13
Start With the Control Boundary, Not the HSM 25
Choose the Deployment and Operating Model 27
Engineer Capacity for Peak and Degraded Operation 29
Govern the Full Key Lifecycle 31
Translate India Localization into Architecture Controls 33
Build Crypto Agility into Long-lived Payment Infrastructure 35
Evaluate Architecture Options with Evidence 36
Where Futurex Helps 38
Choose the Smallest Useful Next Step 41
Conclusion 42
About Futurex 42
References 43

Executive Summary

India's payments story is often best told through its sheer scale. Unified Payments Interface (UPI) alone processed more than 24.5 billion transactions in August 2026[i], while RuPay, Immediate Payment Service (IMPS), National Financial Switch (NFS), National Automated Clearing House (NACH), National Electronic Toll Collection (NETC)/FASTag, Cheque Truncation System (CTS), Aadhaar Enabled Payment System (AePS), Bharat Interface for Money (BHIM), and BHIM Aadhaar support different parts of a payment and clearing ecosystem that now reaches almost every segment of the country's economy.

For technology and security teams responsible for the cryptographic infrastructure behind these services, however, transaction growth is only one part of the architecture problem. A bank can participate in UPI, RuPay, IMPS, NFS, NACH, and CTS simultaneously without having six versions of the same cryptographic requirement.

UPI may place particular pressure on continuous transaction processing availability, participant boundaries, and burst capacity. RuPay introduces considerations for cards, PINs, issuers, acquirers, and acceptance. NFS brings ATM switching and terminal-key operations into the picture, while CTS shifts the discussion toward digital signatures, certificate trust, and cheque-image integrity. NACH has a different operating rhythm since recurring and bulk payment flows create concentrated processing windows, recovery deadlines, and evidence requirements that do not resemble an always-on retail authorization path.

The architecture can also differ between two organizations participating in the same program. One may process transactions in its own data centers, while another relies on a sponsor bank, processor, or managed service for part of the protected path. Differences in transaction volume, key ownership, internal staffing, application design, recovery objectives, and deployment policy can lead to very different answers even when the NPCI program is identical.

THERE IS NO SINGLE "NPCI HSM ARCHITECTURE."

The more useful question is whether, for a specific NPCI program and the organization's role within it, the cryptographic architecture provides the required performance, availability, control, evidence, and operating fit.

Answering that question starts with the service and the transaction path rather than with a hardware security module (HSM) specification. Teams must understand where cryptographic operations actually occur, which participant owns each responsibility, what keys and certificates are involved, how the workload behaves during peak and recovery periods, and what capacity remains when a node, link, site, or maintenance domain is unavailable.

Isolation deserves the same treatment. A workload may justify dedicated physical infrastructure because it has a distinct failure domain, key hierarchy, administrator group, audit boundary, or change cycle. Another may use firmware-isolated virtual HSMs or application-level controls within a carefully designed environment. A managed cloud service may fit a particular workload and responsibility model, while another organization may require direct on-premises control. None of these patterns is inherently the modern answer; the right pattern depends on the workload, the control boundary, and how the organization is prepared to operate it.

Capacity planning becomes more credible once business transaction rates are translated into the cryptographic work actually performed. One transaction may invoke several secure operations, while failures, retries, background activity, and maintenance can add load that is invisible in a simple transaction per second (TPS) calculation. An environment that appears comfortably sized in steady state may behave very differently after a node or site failure, particularly if the surviving infrastructure must absorb peak demand and clear queued work simultaneously. High availability has to be evaluated in terms of usable capacity under failure, not merely as the presence of redundant devices.

India-specific obligations add another layer. Payment data storage, audit logs, key custody, administrative access, backups, disaster recovery, support paths, and cloud-service responsibility are related questions, but they are not interchangeable.

RBI's payment-system data framework, CERT-In's logging requirements, program rules, sectoral obligations, and internal policy need to be mapped to the organization and workload in scope rather than compressed into a generic statement about localization. [ii] [iii] [iv] [v]

Cryptographic change also belongs within the architecture's lifecycle. Payment systems can remain in service for many years while algorithms, certificates, protocols, software dependencies, and standards continue to move. Preparing for post-quantum cryptography (PQC) begins with a cryptographic inventory and dependency map, followed by prioritization, testing, governance, and phased migration, where the risk and use case justify it. PQC should not be presented as an immediate universal requirement across every NPCI program simply because NIST has standardized new algorithms.[vi]

This paper applies that discipline across ten NPCI programs. The profiles do not replace current NPCI operating rules, RBI directions, UIDAI requirements, PCI standards, or an organization's own risk assessment. Instead, they provide a structured way for the payment, infrastructure, security, operations, and risk teams to identify what must be understood before deciding whether to retain, expand, isolate, modernize, or replace an existing environment.

PROGRAM NAVIGATOR

PROGRAM	SERVICE CONTEXT	PRIMARY READER ENTRY POINT
UPI	Real-time account-to-account payments	PSP banks, issuer banks, acquirer banks, TPAPs, PPIs, processors
RuPay	Domestic card issuing and acceptance	Issuers, acquirers, processors, card and ATM teams
NETC/FASTag	Electronic toll collection	Issuer banks, acquirer banks, processors, program technology teams
BHIM	NPCI's UPI-based payment application	Participating banks and platforms supporting BHIM transactions
IMPS	Immediate interbank funds transfer	Banks, PPIs, processors, switching and digital-channel teams
NACH	Recurring and bulk electronic payments	Sponsor banks, destination banks, corporates through banking partners, clearing teams
NFS	Shared ATM switching	Member banks, sponsor banks, sub-members, ATM processors
CTS	Cheque image-based clearing	Presenting and drawee banks, clearing operations, PKI and signing teams
AePS	Aadhaar-enabled bank-led transactions	Issuer banks, acquirer banks, sponsor banks, processors
BHIM Aadhaar	Aadhaar-authenticated merchant payments	Acquirer banks, issuer banks, processors, authentication and settlement teams

A sound evaluation does not have to end in replacement. An organization may conclude that its current environment remains fit for purpose, or that the real need is additional capacity, stronger recovery testing, cleaner key ownership, better workload isolation, or a managed deployment for a defined use case.

One Ecosystem, Ten Distinct Cryptographic Environments

While NPCI provides the common infrastructure and interoperability behind a broad portfolio of Indian payment and settlement services, the umbrella does not make the underlying transaction paths uniform.

UPI and IMPS support immediate account-to-account transfers; RuPay carries card issuing and acquiring responsibilities; NFS connects ATM environments; NACH supports recurring and bulk payments; CTS replaces the physical movement of cheques with electronic images and related clearing data; NETC/FASTag spans issuer, acquirer, acceptance, and settlement processes; and Aadhaar-enabled services introduce another set of institutional and authentication boundaries.

While treating all of this as a single category called “NPCI security” is administratively convenient, it can obscure the architectural decisions that matter. The protected operation varies from one program to another; ownership shifts with the participant role; failure has different business consequences; and the applicable assurance basis can span RBI directions, NPCI operating rules, PCI standards, UIDAI requirements, contractual obligations, and the organization's own control framework.

THE CRYPTOGRAPHIC OPERATION CHANGES

A business transaction is not a cryptographic unit of measurement. Depending on the program, participant role, and implementation, a transaction path may involve PIN processing, message authentication, signing, verification, certificate operations, key wrapping, encryption, or other protected functions.

Some components in the flow may not invoke an HSM at all, while others may generate multiple secure operations for the same customer transaction.

The architecture team needs to identify the secure operations that reach the environment being sized, the mix of those operations, the associated keys and algorithms, how applications open and reuse sessions, and the load generated by retries, reversals, batch activity, key rotation, and recovery. Without that translation, the HSM can be either overbuilt around an ecosystem headline or underbuilt around an average that says little about the actual peak.

OWNERSHIP CHANGES WITH THE PARTICIPANT ROLE

NPCI participation does not mean that every participant owns the same part of the transaction. UPI makes this especially visible because a TPAP, PSP bank, remitter bank, beneficiary bank, PPI, and processor can occupy different positions in the payment flow. The customer may associate the transaction with the application on the screen, while significant authentication, authorization, key management, and cryptographic responsibilities sit elsewhere in the ecosystem.

The same principle applies across card, ATM, and Aadhaar-enabled programs, where participant roles can substantially shift the cryptographic boundary. A RuPay issuer and acquirer perform different functions, while an NFS sub-member may depend on a sponsor or processor for activities that a direct member operates itself.

Business correspondents and merchants can be integral to the customer journey without being the organizations that buy or operate the cryptographic infrastructure, which is why participant directories prioritize relevance over HSM ownership, deployment count, or buying authority.

AVAILABILITY DEPENDS ON THE SERVICE RHYTHM

An immediate payment rail and a clearing workload can both be critical, yet the architecture behind each one may be stressed in very different ways. For real-time services, availability can depend on maintaining low-latency processing during demand bursts while preserving sufficient headroom to withstand a component failure. A batch or clearing environment may be less sensitive to a single transaction's response time but far more sensitive to submission windows, backlog clearance, retries, and reconciliation deadlines.

Distributed ATM and assisted-payment environments introduce another dimension because central HSM availability is only part of the operating picture. Device or terminal keys, remote operations, connectivity, sponsor dependencies, and recovery procedures all influence whether service can be restored safely. An active-active label or an HA pair says very little until the team defines what workload remains after failure, how clients reconnect, whether key state and policy remain correct, and how quickly normal operating capacity can be restored.

THE ASSURANCE BASIS CHANGES

Indian payment infrastructure spans several layers of regulation, scheme governance, technical standards, and internal policy, each applying to a different scope. RBI directions follow the regulated entity and activity; NPCI rules follow the relevant program and participant role; PCI standards can be directly relevant to card and PIN environments without applying in the same way to every other NPCI service. Aadhaar-enabled transactions add UIDAI requirements, bringing the authentication architecture into scope.

Keeping those categories separate prevents architecture recommendations from acquiring the language of compliance without a clear source. A regulator can impose an obligation; a scheme can impose a program requirement; a standard can define a control or validation boundary; and an architect can recommend a design because it improves resilience, isolation, or supportability. Product capability can support those requirements, but an HSM validation or a regional deployment choice does not make the surrounding service automatically compliant.

THE CHANGE BOUNDARY CHANGES

A cryptographic change in one card authorization stack can have a very different testing and rollback surface from a change to a shared enterprise service.

CTS may bring certificate chains, signing applications, and trust anchors into a migration. At the same time, NFS can involve PIN and terminal-key workflows, and UPI can involve a different set of participant interfaces and availability constraints. The broader the dependency graph, the more important it becomes to understand which applications, teams, keys, and recovery procedures move together.

This is also why consolidation and separation should remain design choices rather than campaign conclusions. Consolidation can reduce unnecessary sprawl when workloads genuinely share an operating model and can be isolated appropriately, while dedicated infrastructure can create clearer capacity, failure, administration, or audit boundaries where those distinctions matter.

The right answer depends on the program and the organization, not on a generic preference for shared or dedicated architecture.

Service context	Programs	Questions that usually come forward
Real-time payments	UPI, BHIM, IMPS	Participant role, transaction path, secure-operation rate, latency, high availability, recovery
Cards and acceptance	RuPay, NETC/FASTag	Issuer/acquirer boundaries, payment and key operations, acceptance infrastructure, settlement
ATM switching	NFS	PIN and terminal key controls, switch availability, member/sponsor boundaries, distributed operations
Recurring and clearing	NACH, CTS	Batch or clearing windows, signing and integrity, retries, evidence, backlog recovery
Aadhaar-enabled payments	AePS, BHIM Aadhaar	Issuer/acquirer responsibility, authentication boundaries, device and processor trust

The taxonomy is only a starting point. Before discussing a target architecture, an organization still needs to identify the exact program, its participant role, the applications and processors in scope, the data and key classes involved, the expected workload, the required recovery behavior, and the teams that own the result.

One Organization Can Play Several NPCI Roles

The breadth of NPCI participation inside an Indian financial institution can create a misleading impression of architectural uniformity.

A large public-sector or private bank may support UPI, issue and acquire RuPay cards, participate in IMPS and NACH, operate ATM services through NFS, process CTS clearing, and support Aadhaar-enabled transactions. From an enterprise portfolio perspective, these activities belong to a single organization; operationally, they may span different teams, applications, processors, release cycles, data centers, and control owners.

That separation is not merely organizational. The UPI platform team may own a high-volume digital-payment stack, while card issuing and acquiring sit elsewhere. NFS may be tied to ATM and channel infrastructure; CTS may involve clearing operations and PKI teams; and HSM administration may be centralized under security, even though individual payment teams own the applications that invoke the service.

Large multi-rail banks: Complexity is usually organizational as well as technical

Large banks rarely suffer from an absence of cryptography. More often, they have accumulated multiple HSMs, key management processes, data centers, application owners, vendors, and long-established payment environments, each built at different times for different purposes.

The architecture question is less about whether these organizations need cryptography and more about where common infrastructure genuinely improves operations without blurring ownership, failure domains, or change boundaries.

One program may justify dedicated capacity because the cost of contention or a failed change is unacceptable. Another may be able to use firmware-isolated virtual HSMs inside a program-specific environment. A third may remain on an incumbent platform because it already meets the measured requirement. Central management or monitoring can still be valuable across such a mixed estate, meaning that broader visibility does not necessarily imply common hosting.

Mid-sized and specialized banks: Make the responsibility model visible

Mid-sized commercial banks, small finance banks, payments banks, and other specialized financial institutions may participate in fewer programs but depend more heavily on processors, service providers, or sponsor arrangements. Once those external parties enter the transaction path, key ownership and operational responsibility become more important than the deployment label itself.

The organization needs to know who generates and owns the relevant keys, who administers the HSM, who can approve sensitive key ceremonies or recovery actions, who holds backups, which party produces audit records, and what happens if the provider, network path, or primary site becomes unavailable.

A managed or hybrid model can be entirely appropriate when the program and policy permit it, but outsourcing infrastructure does not transfer the regulated entity's accountability unless the applicable requirement and contract explicitly say otherwise.

Regional, cooperative, and rural institutions: Start with the actual operating boundary

Regional rural banks, cooperative institutions, and smaller participants may access NPCI programs directly, through sponsor banks, or as sub-members.
Their customer reach can be extensive even when specialist cryptographic staffing is limited, which makes it risky to assume either that every participant needs its own dedicated HSM estate or that a managed model is automatically preferable.

Discovery should establish where cryptographic processing takes place, which party controls the keys, what the organization can administer directly, how it obtains audit and incident records, and which recovery actions remain available if the sponsor or provider is unavailable. Those answers may support a compact on-premises design in one organization, a managed service in another, or a sponsor-led model in which much of the cryptographic boundary lies outside the participant's own infrastructure.

Fintechs, TPAPs, PPIs, and processors: Follow the transaction before assigning the HSM

Fintech and payment-technology environments can change quickly, particularly in environments involving cloud-native applications, APIs, and campaign-driven demand.

That speed does not mean the fintech layer bears all cryptographic responsibilities in the transaction. A TPAP should not be assumed to perform bank-side UPI functions simply because the customer experiences the payment through its application, while a processor serving several organizations may operate substantial cryptographic infrastructure and require strong separation between client environments.

The transaction should be traced across organizational boundaries before a product discussion begins. Teams need to identify where credentials are captured, which component authenticates the participant or application, where message integrity is established, which system makes the authorization decision, where the keys reside, and who owns incident response when the service fails.

Only then is it possible to decide whether the fintech or processor layer needs dedicated HSM capacity, firmware-isolated virtual HSMs, a managed service, stronger controls around an existing environment, or no HSM change at that layer at all.

Who needs to be in the room?

Cryptographic architecture decisions rarely belong to one owner because each stakeholder sees a different part of the same service.

The program or payment owner understands the transaction flow and business commitments; infrastructure owns integration, capacity, networking, and service management; security architecture defines trust boundaries and key custody; HSM and key management operations own ceremonies, administration, backup, and recovery; risk and compliance teams interpret obligations; and procurement tests support, lifecycle cost, and exit assumptions.

A useful working session can begin with five facts rather than a long questionnaire: the program and participant role, the protected path, the normal and peak workload, the control owners, and the decision trigger.

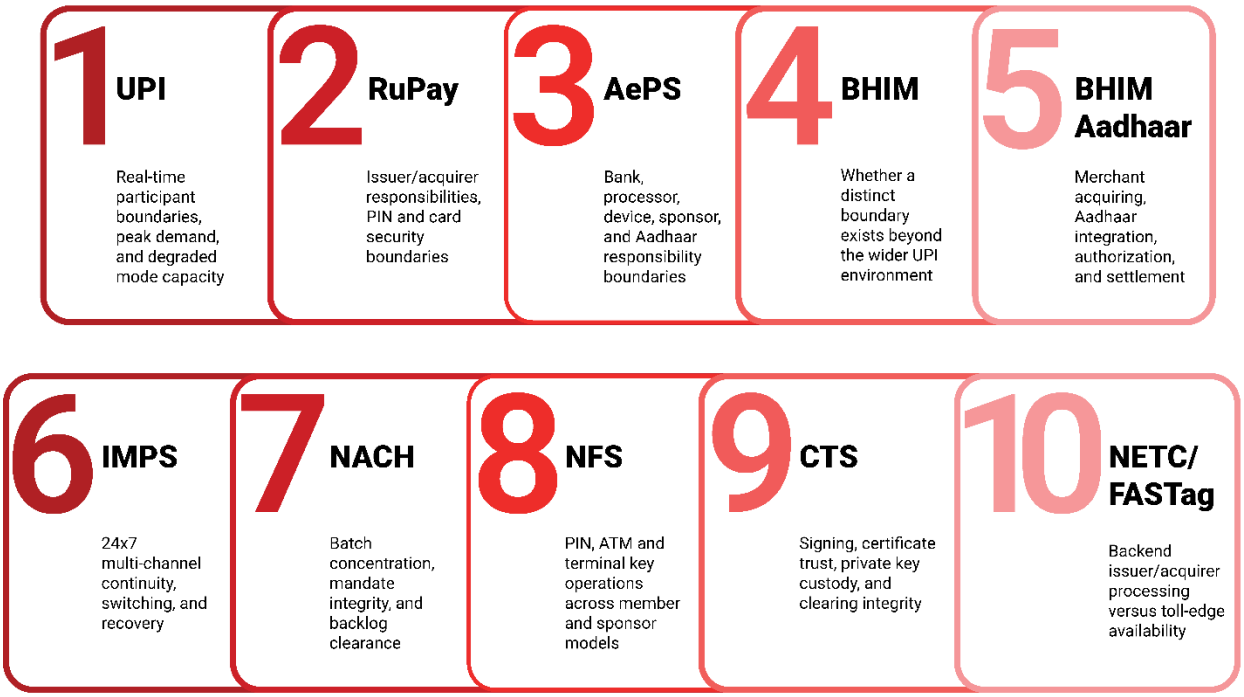
That trigger may be growth, resilience, a refresh, a new program, a control gap, a cloud evaluation, an algorithm change, or simply the need to establish a baseline. The current environment may prove adequate, and the framework should make that conclusion as defensible as a recommendation to change it would be.

FACT TO ESTABLISH	WHAT THE TEAM SHOULD CAPTURE
Program and participant role	Exact NPCI program and the organization's role in the transaction or clearing flow
Protected path	Applications, switches, processors, networks, cryptographic services, data stores, and recovery environments in scope
Workload	Normal load, peak, retries, batch activity, maintenance, latency objectives, and degraded-mode requirement
Control owners	Business, platform, security, HSM operations, risk, compliance, service-provider, and procurement owners
Decision trigger	Growth, refresh, resilience, control gap, cloud evaluation, new program, algorithm change, or baseline assessment

Ten Program-specific Architecture Profiles

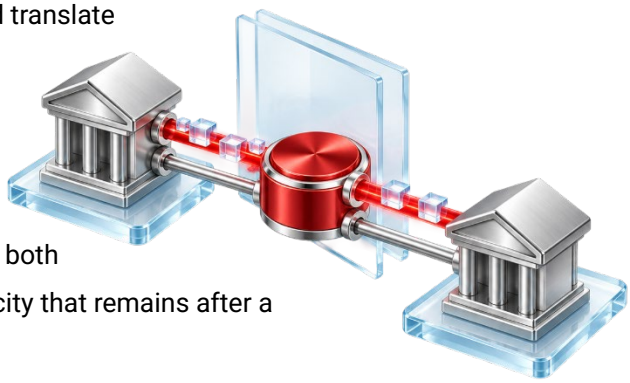
The profiles in this section are discovery guides rather than prescriptive designs.

They identify the participant roles, operating pressures, likely cryptographic touchpoints, and architectural questions to examine for each program, while recognizing that the exact implementation must be checked against current NPCI documentation, applicable RBI and UIDAI requirements, relevant standards, and the organization's own environment.



UPI

UPI supports instant account-to-account payments through a participant model that can include NPCI, PSP banks, remitter and beneficiary banks, TPAPs, PPIs, and processors. NPCI reported more than 24.5 billion UPI transactions in August 2026, illustrating the scale of the ecosystem but not establishing the peak rate or HSM workload for any one participant. For architecture teams, the key starting point is the organization's role, because the cryptographic responsibilities of a PSP bank or issuer differ from those of a TPAP.



A bank or processor sizing the protected part of its UPI path should translate transaction demand into secure operations rather than carry the ecosystem TPS figure directly into the HSM design. Successful payments, failures, retries, reversals, key lifecycle activity, and operational events can create different workloads, while the command mix, session behavior, and network round-trips influence both throughput and latency. Peak periods matter, but so does the capacity that remains after a node, link, site, or maintenance domain is unavailable.

Isolation should follow ownership and failure impact. Separate applications, clients, environments, or institutional tenants may require distinct keys, administrators, policies, and audit boundaries even when they share the same physical estate. Conversely, a TPAP may discover that the bank-side payment function it initially assumed was an HSM requirement at the application layer actually belongs elsewhere in the transaction path.

For large-scale payment workloads, the recommended architecture requires workload-specific validation of the mix of operations, interfaces, concurrency, latency, and degraded-mode capacity.

Architecture Review Questions

- Which participant owns each authentication, authorization, message-integrity, signing, and data-protection function in the path being sized?
- How many secure operations are created by successful, failed, retried, and reversed transactions at the component in scope?
- What peak workload can the surviving environment sustain after a defined failure or maintenance event?

RuPay

RuPay is NPCI's domestic card network, and the cryptographic architecture varies materially depending on whether the organization issues, acquires, processes, or supports acceptance infrastructure.

Card and PIN environments can involve functions such as PIN processing, card verification, EMV-related operations, terminal and issuer key domains, and controlled key exchange. However, those functions must be mapped to the participant role rather than treated as a generic RuPay checklist.



RBI's 2021 Digital Payment Security Controls Directions[vii] are particularly relevant to regulated card environments because they address HSM clustering and backups, access control, application isolation, privileged administration, secure key management, and PIN or CVV-related controls for the entities and activities in scope. These controls provide a useful reminder that a card HSM design is as much an operating model as it is a choice of cryptographic device.

A RuPay review should trace the issuer and acquirer key domains, card and terminal lifecycle, authorization path, HSM command use, key ceremonies, and application connectivity, while also examining whether an outage or maintenance event would force traffic through a weaker or less controlled path. Dedicated infrastructure can create a clear capacity and certification boundary where payment and PIN operations justify it; firmware-isolated virtual HSMs can also support separation inside an applicable environment.

Architecture Review Questions

- Which issuer, acquirer, PIN, card, terminal, and application key domains are actually in scope for this organization?
- Which RBI, PCI, RuPay, and internal controls apply to the role being designed, and where does each one create an HSM requirement?
- Can the environment maintain the required controls and capacity through maintenance, failover, and key-recovery events?

AePS



AePS is a bank-led model that supports interoperable financial and non-financial transactions through business correspondent and micro-ATM environments using Aadhaar authentication.

The cryptographic buyer is the participating or sponsor organization, acquirer, issuer, processor, or relevant technology provider rather than the business correspondent agent or Aadhaar holder.

Architecture discovery should trace the assisted transaction across device, network, processor, bank, NPCI, and Aadhaar authentication boundaries without assuming that every security function belongs to the same system. Teams need to identify who owns device trust, session protection, participant authentication, authorization, settlement, and key or certificate operations, and consult current UIDAI materials before making any specific claims about biometric cryptography or device-level controls.

The distributed channel means that central HSM availability is necessary but insufficient. Device onboarding, network disruption, sponsor dependencies, administrator location, recovery authority, and access to incident records can all determine whether the service can be restored safely.

Where a processor supports several organizations, firmware-isolated virtual HSMs or other supported separation controls may be useful inside the program architecture. That choice should follow a tested and documented capacity and control model rather than a blanket multitenancy proposition.

Architecture Review Questions

- Where do NPCI, bank, processor, device, and UIDAI-controlled responsibilities begin and end in the actual AePS path?
- Which device, certificate, key, administrator, and recovery controls belong to the organization rather than the sponsor or service provider?
- How will the service recover from device, network, sponsor, processor, or central cryptographic failures without weakening the control boundary?

BHIM

BHIM is NPCI's UPI-based payment application, so the first architectural question is whether it creates any cryptographic boundary materially distinct from the organization's wider UPI environment.

For many participating banks and supporting platforms, the relevant transaction path may reuse existing UPI services, keys, connectivity, and recovery processes rather than create an independent HSM estate.



That does not make the profile irrelevant. A bank or processor still needs to establish whether BHIM-related traffic introduces a separate application identity, certificate boundary, capacity allocation, change cycle, or evidence requirement, particularly where supporting platforms or processors maintain a distinct service layer. If no meaningful boundary exists, creating dedicated infrastructure simply to match a program label can add complexity without changing risk.

Architecture Review Questions

- Does BHIM traffic use the same protected bank-side UPI services, or does it create a distinct application, certificate, or operational boundary?
- Would separate infrastructure change risk, capacity, ownership, or recovery, or simply reproduce the existing UPI architecture?

Which bank or processor team owns the cryptographic service that actually supports the BHIM transaction path?

BHIM Aadhaar

BHIM Aadhaar supports Aadhaar-authenticated merchant payments, which creates an acquiring-and-merchant-payment context that should not be collapsed into BHIM or AePS simply because Aadhaar authentication appears in the flow.

The institutional architecture can include acquirer and issuer banks, processors, authentication services, and settlement systems, while merchants and customers remain service users rather than direct HSM buyers.



Discovery should identify which party controls merchant onboarding, device or application trust, Aadhaar authentication integration, transaction authorization, message integrity, settlement, and dispute evidence. It should also establish where sensitive data can appear and which keys or certificates belong to acquiring, issuing, processing, or authentication domains. While shared components with AePS may be entirely reasonable, the overlap needs to be demonstrated through the transaction and control model rather than assumed from the common Aadhaar context.

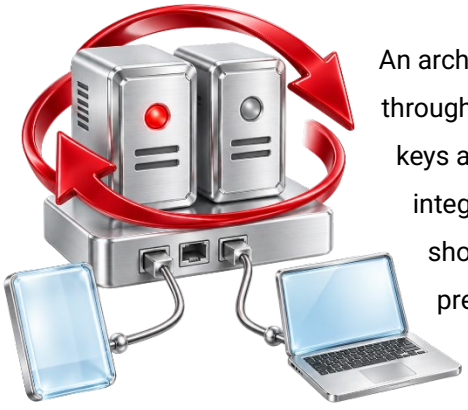
Architecture Review Questions

- What is distinct about the merchant-acquiring, device, authentication, and settlement path compared with AePS?
- Which acquiring, issuing, processor, and authentication domains own the keys and operational evidence?

IMPS

IMPS provides 24x7 interbank funds transfer across channels, including mobile, internet, ATM, branch, and other digital interfaces.

Its channel mix, switching path, participant integration, and legacy dependencies can create a different operational profile even when some bank-side security components are shared.



An architecture review should follow the message from the originating channel through the participating bank or processor and the interbank switch, identifying the keys and protected functions used for participant authentication, message integrity, channel security, and any organization-specific authorization controls. It should also separate channel protection from the central HSM workload to prevent a TLS or application-layer dependency from being mistakenly counted as an HSM operation.

Resilience needs to cover more than online device status. The surviving environment should retain enough secure-operation capacity, preserve the correct key state, and keep administrative actions under control, produce attributable logs, and support reconciliation or backlog clearance after a disruption.

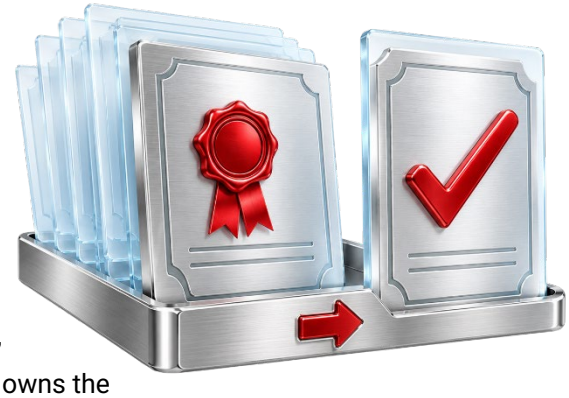
Where an organization has strict ownership or change-control requirements, dedicated capacity can simplify these guarantees; elsewhere, a shared or managed design may be appropriate if isolation and failure behavior remain demonstrable.

Architecture Review Questions

- Which IMPS channels and participant components actually depend on the cryptographic environment being evaluated?
- What capacity and recovery behavior are required after a switch, node, site, or connectivity failure?
- How will the team distinguish central cryptographic constraints from channel, application, or network latency?

NACH

NACH is a useful counterpoint to the assumption that real-time TPS dominates every payment HSM problem.



Recurring and bulk payment flows create a different operating rhythm in which submission windows, mandate processing, file or message integrity, exception handling, settlement, retries, and backlog clearance can matter more than a continuous retail authorization peak.

The protected boundary needs to be determined by the workflow rather than by an average daily transaction count. Teams should identify how mandates and submission artifacts are authenticated, signed, encrypted, archived, revoked, and reconciled; which organization or service provider owns the associated keys and certificates; how corporate-originated instructions enter the bank's controlled environment; and what happens when a batch must be retried or reconstructed after a failure.

A program-specific environment can be valuable where NACH requires distinct signing, verification, message authentication, certificate, or data protection controls. However, the exact cryptographic functions should be verified against the organization's implementation before they are turned into a platform requirement.

Capacity planning should focus on concentrated workload, retry behavior, and the rate at which delayed work must be cleared, because a system that is comfortable at average load can still fail the business objective if it cannot recover before the next processing window.

Architecture Review Questions

- Which mandate, file, message, signing, or certificate operations actually require HSM-backed protection in this implementation?
- What batch concentration, retry, and backlog-clearance rate must the environment support after a disruption?
- Who owns the keys, certificates, recovery artifacts, and evidence across the bank, processor, and corporate boundary?

NFS

NFS connects ATM transactions across member organizations, making PIN security, terminal or host keys, switching availability, sponsor relationships, and remote key operations particularly important areas for architectural discovery.

The party operating the customer-facing ATM does not necessarily own every cryptographic component in the transaction path, so direct members, sponsor banks, sub-members, processors, ATM controllers, and issuer hosts need to be mapped before the HSM boundary is defined.

The review should identify where PIN blocks are processed or translated, where applicable, who provisions and rotates terminal keys, how message-authentication keys are managed, and what happens when a terminal, the key-loading process, a sponsor connection, or a primary cryptographic service fails. While RBI's card and ATM controls are relevant evidence for regulated entities in scope, the actual design still needs to reflect member role, processor dependencies, and the operational impact of recovery.



NFS can justify a dedicated payment HSM infrastructure where PIN, terminal, or switching operations require a tightly controlled boundary and predictable capacity. Firmware-isolated virtual HSMs can also support member, application, or service separation inside an applicable deployment, provided the capacity plan accounts for contention and the administrative model preserves the

required separation. Remote key loading workflows should be mapped to the exact supported process rather than assumed from the presence of distributed ATM infrastructure.

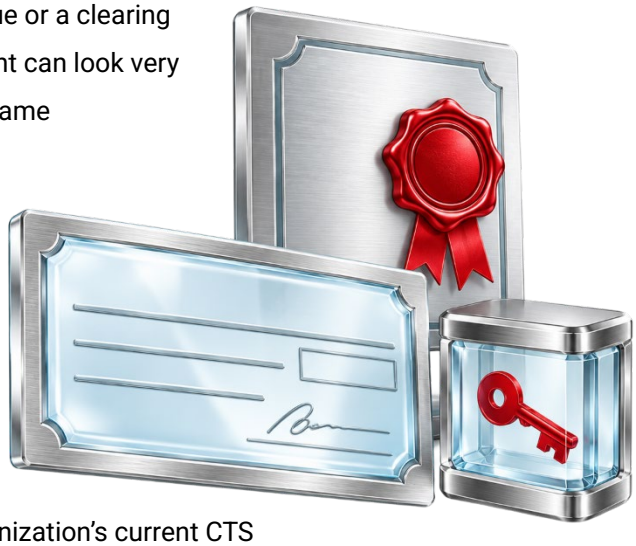
Architecture Review Questions

- Who terminates each protected ATM connection and owns the PIN, terminal, or message-authentication keys in the design?
- How is service restored when a terminal key process, sponsor relationship, processor, or central HSM environment is unavailable?
- Does the isolation model prevent one member, application, or administrative domain from affecting another under peak or failure conditions?

CTS

CTS replaces the physical movement of cheques in the clearing process with electronic cheque images and related data, which moves the cryptographic emphasis away from payment PIN processing and toward authenticity, integrity, signing, certificate trust, private key custody, and dependable evidence across the presenting, clearing, and drawee-bank workflow.

A CTS architecture review should establish the exact signing and verification functions, the certificate authorities and trust anchors involved, the key owners and approval roles, how image and transaction data are protected, and the evidence required when a cheque or a clearing event is disputed. The key lifecycle for a signing environment can look very different from a card or ATM key hierarchy, even when the same organization owns both workloads.



Throughput should also be modeled around clearing windows, image volumes, verification work, retries, and backlog clearance rather than borrowed from a real-time payment profile.

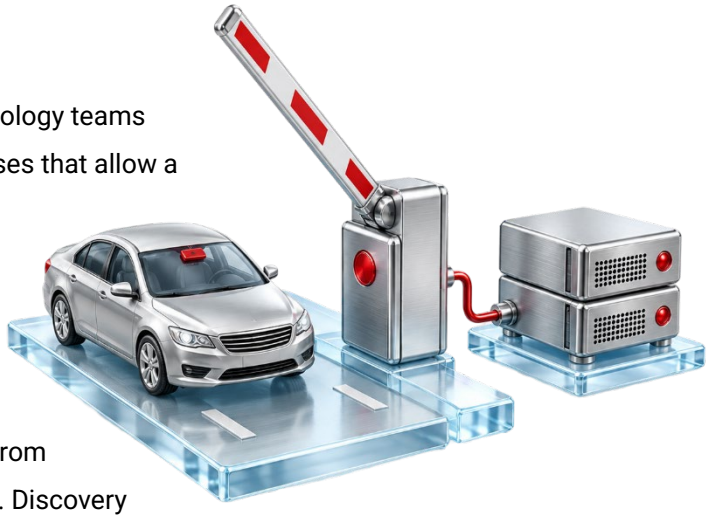
The final design should verify that the algorithm, certificate, interface, and NPCI program-rule details align with the organization's current CTS environment before production approval.

Architecture Review Questions

- Which signing, verification, certificate, and trust anchor functions are performed by the presenting bank, clearing layer, and drawee bank?
- What private key custody, approval, recovery, and certificate lifecycle controls are required for the actual workflow?
- Can the environment process and recover the required clearing workload inside the applicable operating window?
- If components are shared with AePS or another service, has the organization documented the capacity, isolation, and failure implications?

NETC/FASTag

While NETC/FASTag is visible to customers at the road toll lane, the HSM architecture belongs primarily in the institutional processing behind that experience.



Issuer banks, acquirer banks, processors, and program technology teams protect the messages, keys, services, and settlement processes that allow a distributed acceptance environment to operate at a national scale. This distinction matters because toll-plaza response time and central cryptographic latency are not the same engineering problem.

The architecture should separate what happens at the edge from what happens inside issuer, acquirer, and processor systems. Discovery needs to identify the message authentication, certificate, API, key management, reconciliation, and settlement functions that actually reach the protected environment; it should also establish who provisions and rotates the relevant keys, what happens during connectivity disruption, and how exceptional transactions are reconciled when the edge and central systems reconnect.

A dedicated on-premises environment may fit an organization that wants direct control over a stable high-volume processing boundary, while a managed or hybrid model can be considered where the program, risk posture, connectivity, and evidence requirements allow it. The decision should be driven by the institutional role and the actual backend workflow rather than by an assumption that every FASTag transaction produces the same HSM pattern.

Architecture Review Questions

- Which issuer, acquirer, processor, or settlement functions create cryptographic operations in the environment being evaluated?
- How are key ownership and recovery handled when toll, network, processor, or bank connectivity is disrupted?
- Does the target architecture distinguish edge availability from the capacity and recovery requirements of the central cryptographic service?

Start With the Control Boundary, Not the HSM

An HSM protects keys and performs approved cryptographic operations inside a controlled boundary; it does not secure the complete payment application, network, database, endpoint, or operating process around it.

Architecture becomes more defensible when teams define that boundary before they compare products, because the exercise exposes where keys are created, which applications can invoke them, who administers policy, what evidence is produced, and which failures remain outside the HSM's control.

The protected service should be documented end-to-end, including the NPCI program and participant role, the applications and processors in scope, normal and exceptional transaction paths, the data and key classes created or handled, the cryptographic operation performed at each step, the owners who approve policy and recovery, and the regulatory or program requirement behind each control.

Doing this early prevents three common errors: sizing based on total transactions rather than HSM calls, assigning cryptographic responsibility to the wrong participant, and treating product validation as evidence that the deployed service is compliant.

ISOLATION IS A DESIGN DECISION, NOT A PRODUCT FEATURE CHECKBOX

Isolation can be physical, firmware-level, logical, operational, or organizational. While physical separation can create a clear failure and change boundary, it can also add infrastructure and lifecycle work.

The reason for separation should be explicit, because available capacity alone does not prove that workloads belong together. Different legal entities, key owners, administrator groups, release cadences, recovery objectives, certification scopes, or consequences of failure can all justify stronger separation. At the same time, separate hardware should not be treated as inherently superior if the same control objective can be met more simply through a tested and documented virtual or logical design.

ISOLATION PATTERN	STRONG FIT WHEN	QUESTIONS TO ANSWER
Dedicated physical environment	Capacity, failure domain, key ownership, change control, or audit scope requires a clearly separate boundary	Who owns hardware lifecycle, backup, DR, maintenance, and support access?
Firmware-isolated virtual HSM	Several workloads or administrative domains can share a device while retaining supported cryptographic isolation	How are administration, keys, capacity, policy, and failure impact separated?
Application-level controls	Applications inside one approved HSM boundary require finer client, key, or command restrictions	Where are keys, logs, backups, administration, support, and recovery, and who controls each?
Managed service instance	The program and policy permit managed infrastructure with defined customer and provider responsibilities	Where are keys, logs, backups, administration, support, and recovery, and who controls each?

Choose the Deployment and Operating Model

Deployment and isolation are related decisions, but they are not the same decision. An organization can operate dedicated on-premises infrastructure, use firmware-isolated virtual HSMs within an on-premises estate, consume a dedicated managed service, or combine local and managed environments in a hybrid design.

The more useful comparison is not 'hardware versus cloud'; it is who owns and operates each part of the service, where the protected assets and evidence reside, how capacity is added, and how the environment is recovered or exited.

ON-PREMISES

On-premises appliance or PCIe deployment can give an organization direct custody and close integration with payment or clearing applications, which may be attractive for stable high-scale workloads, specific connectivity models, or environments where internal teams already operate mature HSM processes.

That control comes with responsibility for facilities, hardware lifecycle, capacity, firmware governance, ceremonies, monitoring, spares, support access, backups, and disaster recovery, so the architecture should be evaluated during maintenance and site loss rather than only in steady state.

MANAGED CLOUD SERVICES

Managed cloud HSM services reduce the burden of acquiring and operating physical infrastructure and can provide a practical path for defined production, development, recovery, or migration use cases. The operating responsibility does not disappear; it changes shape.

RBI's directions for non-bank payment system operators, together with outsourcing and payment data requirements, make governance, segregation, access, recoverability, audit rights, and provider accountability central to the evaluation of in-scope entities.

HYBRID

Hybrid architecture supports staged migration, development, and testing, disaster recovery, or a deliberately mixed operating model in which different workloads remain in the environment that best fits their control requirements.

The design needs to document how keys are created, wrapped, transferred, restored, and destroyed across the boundary; how policy remains consistent where required; who authorizes failover; and what happens when the connection between environments is unavailable.

On-Premises



Strong fit when: Direct infrastructure control, predictable connectivity, mature internal operations, or dedicated program boundary is important

Customer questions: Facilities, lifecycle, key ceremonies, monitoring, backup, DR, staffing, support

Managed Cloud



Strong fit when: The program permits a managed model and the organization wants defined service responsibilities without operating physical infrastructure

Customer questions: Region, data and key location, provider access, audit rights, connectivity, recovery, SLA, exit

Hybrid



Strong fit when: Migration, DR, testing, or different control requirements justify more than one operating environment

Customer questions: Key movement, policy consistency, dependencies, failover authority, latency, support model

Choose the Deployment and Operating Model

Capacity planning becomes useful only after the team has translated business demand into the secure operations performed by the cryptographic environment.

A transaction can create several HSM calls, while failed requests, retries, reversals, batch jobs, key rotations, device onboarding, and recovery activity can consume capacity outside the successful transaction path.

The command mix, key types, message sizes, connection patterns, concurrency, and latency objectives can also substantially change the performance profile, which is why average transaction volume alone rarely yields a defensible HSM design.

Conceptual sizing model

This is an engineering model rather than a regulatory formula, and each input has to be defined for the program and component being sized.

Required secure-operation capacity = peak transaction rate x cryptographic operations per transaction x degraded-mode factor x headroom factor

Peak transaction rate should be measured at the relevant participant boundary rather than inferred from ecosystem totals. Operations per transaction should include the successful path as well as failures, retries, reversals, and recovery. The degraded-mode factor represents the capacity the service still needs after a node, site, link, or maintenance domain is lost, while headroom accounts for growth, short bursts, background work, and forecast uncertainty without relying on a single universal percentage.

Latency also needs more discipline than the average. P50 can describe ordinary behavior, while P95 and P99 expose the tail that often matters during demand spikes or partial failure. Queue depth, error rate, client reconnection, session behavior, network round-trip times, and the rate at which a backlog can be cleared are often more useful than a maximum throughput figure viewed in isolation. An HSM can be performing within specification while the surrounding application still experiences unacceptable end-to-end delay.

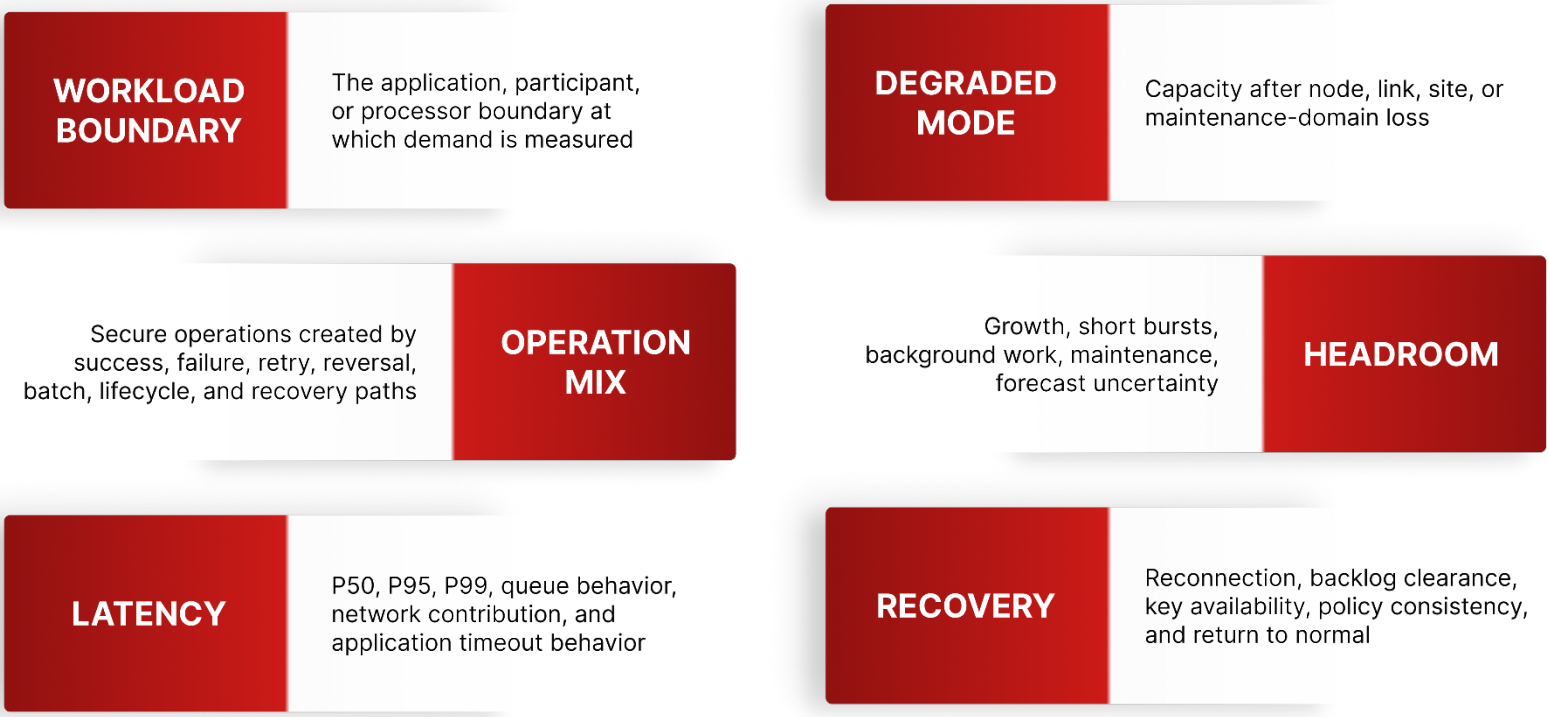
Design for the Failure that Has Already Happened

High availability should be tested as a degraded operating state rather than inferred from topology.

If an active-active pair loses one node, the remaining capacity must still support the required workload; if a site is lost, applications need a known reconnection or rerouting path, key state and policy must remain correct, and the resulting logs must remain attributable and time-consistent. Recovery procedures should also preserve the intended approvals rather than relax controls simply because the organization is under pressure to restore service.

Maintenance deserves the same attention. Firmware changes, key rotations, planned failover exercises, and hardware work can temporarily reduce available capacity even when nothing has failed, which means a design sized only for a perfect steady state can enter a risky operating zone during ordinary maintenance.

The architecture must be tested with production-representative traffic and the target HA or DR topology, including the loss scenarios that the organization is expected to survive.



Choose the Deployment and Operating Model

The strongest HSM in the architecture cannot compensate for an undefined key owner.

Keys pass through a lifecycle that includes creation, custody, use, distribution, rotation, backup, recovery, revocation, and destruction, and weaknesses at any of those stages can undermine the hardware-backed protection around the key itself.

A defensible program architecture assigns ownership and evidence to the lifecycle rather than treating the HSM purchase as the end of the control design.

Build the Inventory Around the Program

The inventory should include only the key and certificate classes that the program actually uses.

Depending on the implementation, that may include local or root master keys, issuer or acquirer keys, PIN or terminal keys, message-authentication keys, signing keys, API and TLS certificates, data encryption keys, wrapping keys, backup keys, and device credentials.

Importing a generic payment key list from another program can be as misleading as omitting a key class that exists only in the local application architecture.

LIFECYCLE QUESTION	DECISION THE ORGANIZATION NEEDS
Ownership	Legal entity, business owner, system owner, risk owner, and operational custodian
Operation Mix	Entropy source, approval, ceremony, dual control where required, and environment
Storage	HSM or protected boundary, backup location, export policy, and recovery artifacts
Use	Applications, commands, clients, algorithms, interfaces, and policy restrictions
Distribution	Wrapping, transport, injection, acknowledgement, and chain of custody where applicable
Rotation	Schedule, event triggers, overlap period, dependent application changes
Recovery	Approvers, restore artifacts, site, test cadence, and emergency restrictions
Retirement	Revocation, destruction, dependent systems, retained records, and proof of completion

Build the Inventory Around the Program

Privileged access should be individually attributable, strongly authenticated, least-privilege, and monitored, while sensitive ceremonies and recovery actions should use the multi-person control required by the applicable policy or standard.

Application identities should be limited to the keys and operations they require, and human administrators should not rely on shared accounts, as this makes later investigation difficult.

Make Recovery Ordinary Enough to be Trusted

Recovery is often where hidden dependencies emerge.

A backup can exist without a tested restore path, key custodians may be unavailable when the procedure is invoked, or a provider may hold an undocumented dependency that becomes visible only during a site loss.

The recovery plan must be rehearsed under realistic scenarios to demonstrate that the cryptographic service can be restored within the required RTO and RPO without bypassing the key controls that make the service trustworthy in normal operation.

Translate India Localization into Architecture Controls

'Data localization' is often used as shorthand for several obligations that sit at different layers of the architecture.

A rigorous review separates payment data storage from log retention, key custody, administrative access, cloud service responsibility, and disaster recovery, because a correct answer to one of those questions does not automatically answer the others.

PAYMENT DATA STORAGE

RBI's payment system data direction requires that payment system data in scope be stored in systems located in India, and its FAQ describes the relevant data as including end-to-end transaction details, customer and payment-sensitive data, payment credentials, and transaction data.

Where payment processing occurs outside India, the FAQ requires that the data be deleted from systems abroad and returned to India within 1 business day or 24 hours of processing, whichever is earlier.[viii]

That requirement does not by itself answer where every cryptographic component must run. HSM clients, management systems, SIEM platforms, support tools, backup media, and recovery environments can all become relevant to that analysis even when the cryptographic key never leaves a protected boundary.

LOGS AND OPERATIONAL EVIDENCE

CERT-In's 2022 directions require covered entities to enable logging across ICT systems, retain them securely for a rolling 180 days, and maintain them within Indian jurisdiction; these directions also establish incident reporting and time synchronization requirements for in-scope entities[ixx]
RBI's 2024 directions for non-bank payment system operators introduce additional audit log requirements for those entities[xi]. Since these obligations have different subjects and scopes, they should not be collapsed into one generic retention statement.

HSM audit events, management actions, application logs, infrastructure logs, and incident records should be classified according to the applicable obligation. The objective is not merely to retain data; it is to

preserve enough context, integrity, time information, and access control for the organization to explain what happened when an incident, audit, or dispute requires reconstruction.

CLOUD AND OUTSOURCING

RBI does not treat all cloud use as prohibited; instead, the relevant directions emphasize governance, segregation, access control, recoverability, independent review, contractual responsibility, and payment data storage for the entities in scope.

A managed service should be evaluated based on the architecture and contract rather than on assumptions about the cloud as a category.

For managed cloud services, the organization should document where keys, backups, logs, management functions, support access, and recovery operations occur; which responsibilities remain with the customer; and which service, redundancy, connectivity, and failover assumptions underpin the design.

CONTROL OBJECT	LOCATION QUESTION	OWNERSHIP QUESTION
Payment Transaction Data	Legal entity, business owner, system owner, risk owner, and operational custodian	Who creates, accesses, retains, deletes, and proves integrity?
Cryptographic Keys	Where are keys, wrapping keys, backups, and recovery artifacts?	Who controls creation, use, restore, rotation, and destruction?
HSM And Management Logs	Where are events generated, exported, retained, and reviewed?	Who can read, change, suppress, or delete them?
Administration	From which locations and devices can privileged users connect?	Who approves, authenticates, monitors, and revokes access?
Support	Can a provider access systems, metadata, logs, or data across borders?	What do the contract, architecture, and emergency process permit?
Backups And DR	Where are data and key backups stored and tested?	Who can authorize and execute recovery?

Build Crypto Agility into Long-lived Payment Infrastructure

Payment infrastructure can remain in service for many years while the algorithms, certificates, interfaces, and standards around it continue to change. Crypto agility is the ability to identify those dependencies, govern them, test alternatives, operate controlled transition states, and retire old mechanisms without creating an uncontrolled service disruption. The practical starting point is an inventory, not a migration deadline.

NIST finalized FIPS 203 for ML-KEM, FIPS 204 for ML-DSA, and FIPS 205 for SLH-DSA in August 2024. While those standards matter to long-term cryptographic planning, they do not impose a universal NPCI or RBI deadline for replacing all current payment mechanisms. Prioritization should reflect data lifetime, public-key dependence, trust-anchor lifetime, application upgrade difficulty, external counterparties, and the consequences of compromise.

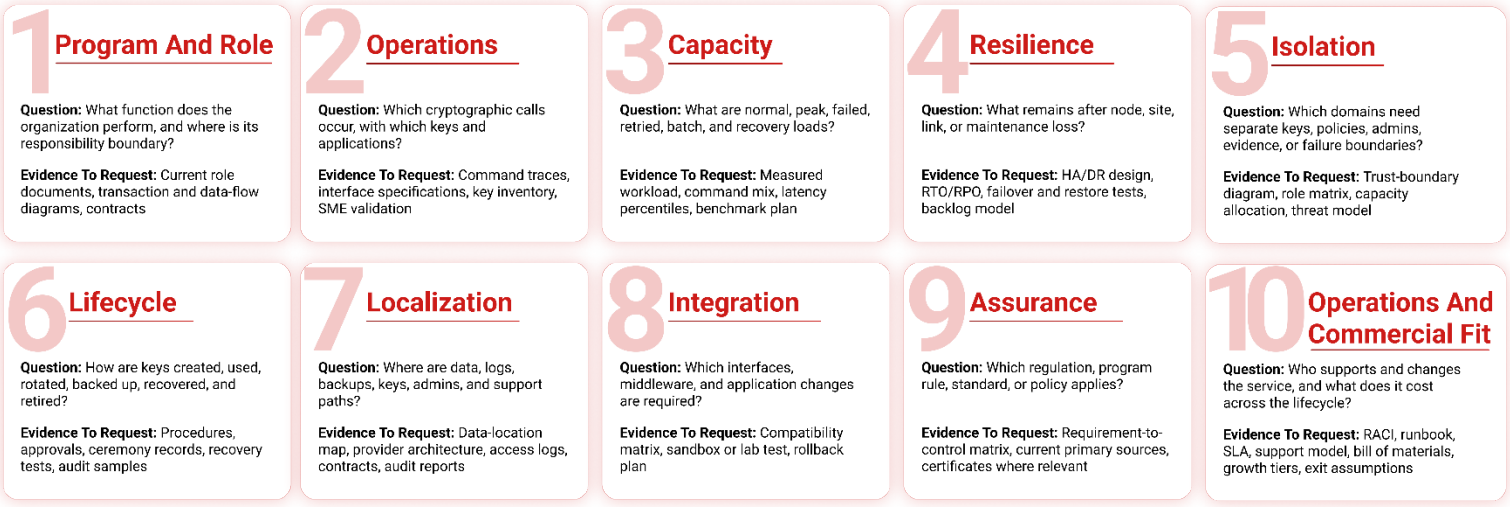
The inventory should cover algorithms, key sizes, certificates, protocols, libraries, HSM firmware, application commands, middleware, public-key uses, long-lived sensitive data, and systems that are difficult to update. Teams can then test new algorithms in isolated environments, define hybrid or phased patterns where appropriate, and make application dependencies less brittle before a mandatory change arrives.

Evaluate Architecture Options with Evidence

A useful architecture review compares credible options rather than assuming that a replacement is the only acceptable outcome.

Depending on the program and current state, the organization may retain the existing environment, add capacity, correct an operational gap, isolate a workload, use firmware-isolated virtual HSMs inside a program-specific boundary, introduce a managed service, create a hybrid design, or replace an incumbent platform.

The evidence required to choose among those options should be visible before the product decision is made.



THREE DEFENSIBLE OUTCOMES

A mature review usually leads to one of three positions.

1. Proceed because the program, role, control boundary, capacity, resilience, key ownership, and validation plan are sufficiently defined for detailed design.

OR

2. Proceed after gap closure because the direction appears sound but missing evidence or controls, such as uncertain provider access, untested recovery, or insufficient failure-mode capacity, still need to be resolved.

OR

3. Escalate for program, legal, compliance, or risk interpretation because an obligation, participant responsibility, card or PIN boundary, Aadhaar requirement, or cross-border access path remains unclear.

The option to make no material change should remain visible throughout. If the current environment meets the measured requirements for capacity, resilience, lifecycle, control, and cost, retaining it with targeted improvements may be the most defensible outcome. That neutrality gives decision makers a clearer reason to act when the evidence does support change.

Where Futurex Helps

Futurex's wide range of cryptographic solutions becomes helpful after an organization has defined the architecture problem.

Futurex's Excrypt HSMs provide the hardware-backed cryptographic foundation; CryptoHub unifies management, policy, automation, monitoring, and lifecycle control across supported cryptographic services; VirtuCrypt delivers Futurex capabilities through managed cloud services; and HubIQ provides dashboards, reporting, notifications, alerts, and operational visibility across supported Futurex workloads.

Keeping those roles separate prevents management, monitoring, and cryptographic processing from being collapsed into one generic platform claim.

REQUIREMENT	FUTUREX SOLUTION	PROOF BEFORE COMMITMENT
Hardware-Backed Payment Or General-Purpose Cryptography	Futurex Excrypt HSMs	Exact operations, interfaces, key model, applicable FIPS 140-3 Level 3 compliant configuration, PCI HSM validation scope, workload-specific test
High Secure-Operation Demand	Excrypt HSM performance up to the approved workload maximum	Command mix, concurrency, topology, latency, degraded-mode capacity
Workload Or Administrative Isolation	Firmware-isolated virtual HSMs and supported isolation controls	Administration model, key separation, capacity allocation, failure analysis
Unified Cryptographic Management And Orchestration	CryptoHub	Supported workflows, integrations, roles, policy, APIs, reporting, operating design
Managed Cloud Cryptography	VirtuCrypt	Regional availability, residency design, access, key custody, logs, connectivity, SLA, recovery, exit
Operational Monitoring And Intelligence	HubIQ	Supported telemetry, alerting, reporting, retention, access, operations integration
PQC Preparedness	ML-KEM and ML-DSA support on applicable Futurex platforms	Model, firmware, validation boundary, interface, interoperability, migration test

EXCRYPT HSMS

Futurex's Excrypt HSMs deliver payment and general-purpose cryptography through one native HSM architecture, with hardware-backed key protection and deployment flexibility. Applicable configurations are FIPS 140-3 Level 3 compliant, with PCI HSM validation tracked as a separate, workload-specific claim. Performance and isolation depth vary by configuration, up to 100,000 payment operations per second and up to 75 firmware-isolated virtual HSMs in approved deployments.



For NPCI use cases, the relevant value may be payment command support, signing, general-purpose cryptography, isolation, or available performance headroom, depending on the program.

CRYPTOHUB

CryptoHub is Futurex's unified cryptographic management and orchestration platform that integrates HSM operations, enterprise key management, PKI and CA, data protection, policy, automation, monitoring, and lifecycle control into a single operating model.



Within the NPCI campaign, that capability is relevant when a defined program or broader cryptographic estate has a management or governance requirement.

VIRTUCRYPT

VirtuCrypt is Futurex's managed cloud services portfolio that delivers cloud payment HSMs, key management, and CryptoHub cloud capabilities through globally distributed infrastructure.

A managed deployment like VirtuCrypt is suitable for production, development, recovery, migration, or other approved workloads when the proposed regional service, customer responsibilities, key and data location, connectivity, support access, and recovery architecture meet the organization's requirements.

HUBIQ

HubIQ provides monitoring and operational intelligence through supported dashboards, reporting, notifications, alerts, and workload visibility. It strengthens capacity monitoring, health assessment, operational response, and reporting where supported, while remaining separate from the HSM, key management engine, policy engine, or cryptographic processing layer.

An organization can evaluate dedicated or isolated cryptographic foundations for specific NPCI workloads, use common management or monitoring where that adds operational value, and adopt on-premises, managed cloud, or hybrid deployment patterns according to the program and responsibility model.

Choose the Smallest Useful Next Step

The right next step depends on how much the organization already knows. A team still mapping its responsibilities may begin with the program-specific brief and use the architecture questions to establish a baseline.

An organization with a defined workload can use a readiness checklist or discovery worksheet to document the current environment, capacity, isolation, key lifecycle, recovery, localization, and change constraints before it involves a solutions team.

A qualified architecture workshop becomes useful when the organization has a defined NPCI program or modernization initiative, an identified owner, a current-state problem or requirement, credible timing, and an agreed purpose for the discussion.

The discussion should begin with the program, participant role, current architecture, workload, recovery requirements, and control constraints; the output should be an agreed-upon architecture direction or next technical action, rather than a generic compliance determination or an automatic recommendation to replace the incumbent platform.

Still mapping the program and responsibilities?	NEXT STEPS	Use the relevant program brief and define the participant and control boundary
Current state is understood but the architecture question is open?	NEXT STEPS	Complete the readiness or workload discovery checklist and identify evidence gaps
Defined initiative, owner, scope, and timing?	NEXT STEPS	Request a qualified architecture workshop to agree the next technical validation or design action

Conclusion

NPCI is a single ecosystem, not a single cryptographic workload. The organization that begins with a generic 'NPCI HSM' requirement risks sizing the wrong boundary, assigning controls to the wrong participant, or choosing an operating model that becomes difficult to recover, audit, or change.

The organization that starts with the program, participant role, transaction path, key lifecycle, and failure model can make a far more defensible decision about what should remain dedicated, what can be isolated virtually, where managed infrastructure fits, and where no change is necessary.

That discipline applies whether the immediate concern is UPI growth, RuPay or NFS payment security, NACH batch integrity, CTS signing, NETC issuer and acquirer processing, or Aadhaar-enabled payments. Capacity needs to be translated into secure operations; resilience needs to be proven under failure; isolation needs a stated purpose; and India localization needs to be traced across data, keys, logs, administrators, support, backups, and recovery, rather than reduced to a data center location.

Futurex supports those program-specific architectures through Excrypt HSMs, CryptoHub, VirtuCrypt, and HubIQ, using only the components the defined use case actually requires. The design question should remain the same: for this NPCI program and for our role in it, which cryptographic architecture provides the required performance, availability, control, evidence, and operating fit?

ABOUT FUTUREX

For over 40 years, Futurex has been an award-winning leader and innovator in the encryption market, delivering uncompromising enterprise-grade data security solutions. Over 15,000 organizations worldwide trust Futurex to provide groundbreaking PQC-ready hardware security modules, key management, PKI and CA, data protection, and cloud HSM solutions.

Futurex is headquartered outside of San Antonio, Texas, with regional offices worldwide and over a dozen data centers across five continents. Futurex delivers unmatched support for its clients' mission-critical data encryption and key management requirements.

References

[i] NPCI, UPI Product Statistics, reporting period August 2026, https://www.npci.org.in/product/ecosystem-statistics/upi
[ii] Reserve Bank of India, Master Direction on Digital Payment Security Controls, February 18, 2021, https://www.rbi.org.in/Scripts/NotificationUser.aspx?Id=12032&Mode=0
[iii] Reserve Bank of India, Master Directions on Cyber Resilience and Digital Payment Security Controls for non-bank Payment System Operators, July 30, 2024, https://www.rbi.org.in/Scripts/NotificationUser.aspx?Id=12715&Mode=0
[iv] Reserve Bank of India, Storage of Payment System Data directive and FAQ, April 6, 2018 and June 26, 2019, https://www.rbi.org.in/commonperson/english/Scripts/FAQs.aspx
[v] Indian Computer Emergency Response Team, Directions relating to information security practices, procedure, prevention, response and reporting of cyber incidents, April 28, 2022, https://www.cert-in.org.in/PDF/CERT-In_Directions_70B_28.04.2022.pdf
[vi] NIST, NIST Releases First 3 Finalized Post-Quantum Encryption Standards, August 13, 2024, https://www.nist.gov/news-events/news/2024/08/13/nist-releases-first-3-finalized-post-quantum-encryption-standards
[vii] Reserve Bank of India, Master Direction on Digital Payment Security Controls, February 18, 2021, https://www.rbi.org.in/Scripts/NotificationUser.aspx?Id=12032&Mode=0
[viii] Reserve Bank of India, Storage of Payment System Data directive and FAQ, April 6, 2018 and June 26, 2019, https://www.rbi.org.in/commonperson/english/Scripts/FAQs.aspx

[x] Indian Computer Emergency Response Team, Directions relating to information security practices, procedure, prevention, response and reporting of cyber incidents, April 28, 2022, https://www.cert-in.org.in/PDF/CERT-In_Directions_70B_28.04.2022.pdf
[xi] Reserve Bank of India, Master Directions on Cyber Resilience and Digital Payment Security Controls for non-bank Payment System Operators, July 30, 2024, https://www.rbi.org.in/Scripts/NotificationUser.aspx?Id=12715&Mode=0