PAYMENTS · DATA · AI · COMPLIANCE

ACH Payments Delivery and Support Runbook

From requirement gathering to live L1, L2 and L3 support: phases, exit criteria, sign-off, penny testing, compliance, security, alerts and incident runbooks.

SP Sankar · Payment Product Consultant

sp@paymentconsultant.top · paymentconsultant.top · linkedin.com/in/spsankar

Version 2026 · Always confirm current rules with the latest Nacha Operating Rules and your ODFI or sponsor bank.

Contents

1. Purpose and scope
2. Planning vs delivery alignment
3. ACH basics for the team
4. Phase 1: Requirement gathering
5. Phase 2: Design
6. Phase 3: Development
7. Phase 4: Unit testing
8. Phase 5: Performance testing
9. Phase 6: End-to-end testing
10. Phase 7: UAT
11. Phase 8: Release management
12. Phase 9: Deployment and cutover
13. Compliance
14. Internal and external audit
15. Open source software (OSS)
16. Password and access policy
17. Server and certificate management
18. Software certification and licences
19. Production support model: L1, L2, L3
20. Operational runbooks
21. Disaster recovery (DR) and business continuity
22. Notifications and alerts
23. Monitoring, KPIs and reporting
24. Checklists and RACI
25. Appendix: return codes and glossary
26. Sources

Purpose and scope

This runbook takes an ACH payments change from first requirement to live production support, with the checks, owners and evidence needed at every step. Use it for a new ACH product, a new SEC code, a processor migration or any change to how your bank or fintech originates or receives ACH.

Who uses it: product owners, business analysts, architects, developers, QA, performance engineers, release managers, operations, compliance, information security and the L1, L2 and L3 support teams.

How it is organised: nine delivery phases (requirements to deployment), which overlap in sprints, then the controls that run across every phase (compliance, open source software, passwords and access), then running the service (support model, incident runbooks, monitoring), then checklists and reference tables.

Golden rules

1. Phases overlap in practice: teams work in sprints, and design, build and testing run side by side for each story. Only a few gates are hard stops before production (see below).
2. Every requirement traces to a test, and every test result is kept as audit evidence.
3. Money movement is never tested with real customer accounts outside production; production checks use company-owned accounts and tiny amounts.
4. Anything that can move money twice must be prevented by design, not by procedure.

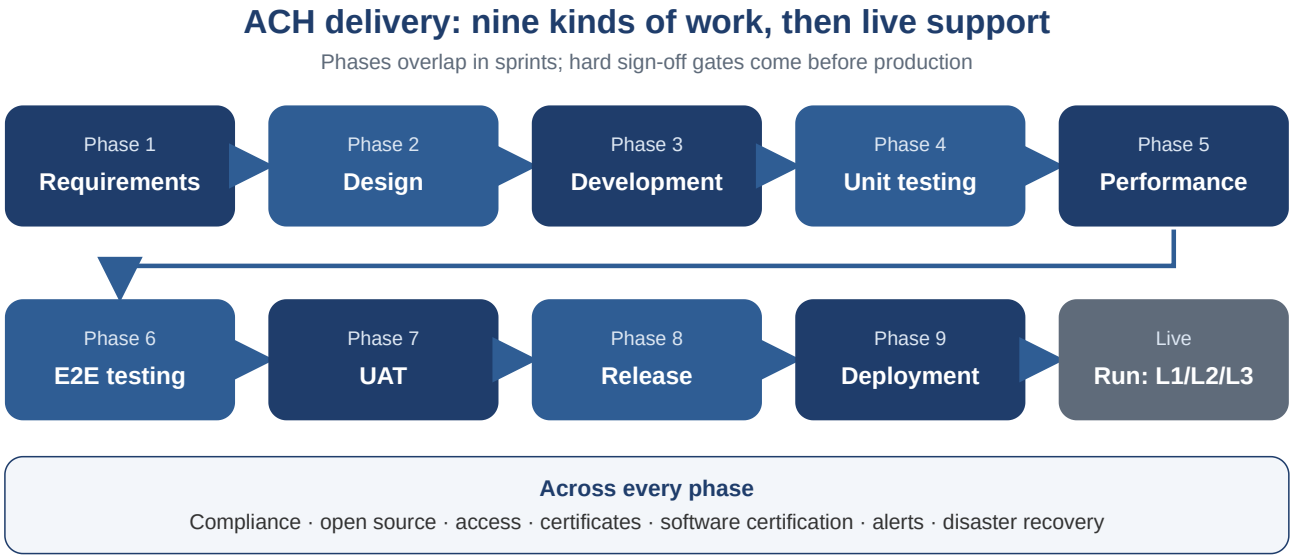
How this fits Agile

The nine phases describe the *work* that has to happen, not a waterfall schedule. In a Scrum or SAFe team, one sprint can hold requirements refinement for the next stories, development and unit testing of current stories, and E2E testing of the last increment, all at once.

What	How it works in Agile
Story level	Definition of Ready before a story starts; Definition of Done (code reviewed, unit tested, automated tests passing) before it closes
Sprint level	Sprint review with the product owner; performance and security checks automated in the pipeline every sprint
Increment level	E2E and UAT run on each releasable increment, often in parallel with the next sprint's build
Hard gates before production	E2E sign-off, UAT sign-off, bank certification (when connections or formats change), CAB go/no-go, and passing penny tests

In short: overlap the work, but never skip the hard gates. Money can't be un-sent, and regulators and sponsor banks expect the sign-off evidence whichever method you use.

Assumptions: US ACH Network under Nacha Rules, with your bank acting as an ODFI (sending bank), an RDFI (receiving bank), or both. Fintechs working through a sponsor bank follow the same steps, with the sponsor bank's rules on top. Times are Eastern Time (ET).



ACH delivery lifecycle

> **How to read this picture:** Read it left to right, then down: the boxes show the order work generally flows, though in sprints several run at once. The shaded band lists the controls that apply in every phase.

Planning vs delivery alignment

A plan is only useful if delivery follows it, and delivery is only useful if it delivers what the plan promised. In ACH, some dates can't move: bank certification windows, Nacha rule effective dates, Federal Reserve holidays and change freezes (month end, year end). So alignment means planning around fixed dates and checking every sprint that delivery is on track.

How plans flow into delivery

Level	What it holds	Who owns it	Reviewed
Product roadmap	Business outcomes, regulatory and bank dates, big features	Product manager	Quarterly
Release or PI plan	Features committed for the quarter, dependencies, risks	Product manager, delivery lead	Every planning increment (8–12 weeks)

Level	What it holds	Who owns it	Reviewed
Sprint plan	User stories the team commits to for 2 weeks	Product owner and team	Every sprint
Release calendar	Deployment dates, freezes, bank cutover windows	Release manager	Weekly

Practices that keep them aligned

- **One backlog, fully traced:** every requirement links to its stories, tests, defects and release, so you can see what is planned, built, tested and live.
- **Definition of Ready and Definition of Done:** a story doesn't enter a sprint without clear acceptance criteria and test data; it isn't done until tested, reviewed and documented.
- **Plan to real capacity:** use the team's actual velocity, minus holidays, production support time and audit work. Don't plan to 100%; keep 15–20% buffer for support and surprises.
- **Fixed dates first:** put bank certification, Nacha rule dates and freezes on the calendar before planning features, and work backwards from them.
- **Dependency board:** track what you need from the bank, vendors and other teams, with owners and dates.
- **RAID log:** risks, assumptions, issues and dependencies, reviewed weekly.
- **Change control for scope:** new work enters through the product owner, and something of equal size comes out, or the date moves, by decision, not by accident.
- **Regular checkpoints:** sprint review and demo every sprint, a mid-quarter health check, and a planning increment review at the end.

Measures of alignment

Measure	What it shows	Healthy sign
Say/do ratio	Committed stories delivered in the sprint	80% or more, steady
Feature predictability	Planned features delivered in the quarter	80% or more
Scope change	Stories added or removed after sprint start	Low, and always agreed
Milestone status	Fixed-date milestones on track	All green or with an agreed recovery plan
Carry-over	Stories rolled to the next sprint	Small, and not the same ones twice

Measure	What it shows	Healthy sign
Defect escape rate	Defects found after release	Falling over time

Warning signs and what to do

Warning sign	Likely cause	What to do
Say/do ratio below 70% for two sprints	Over-planning, hidden support work, poor story readiness	Re-plan to real velocity; enforce Definition of Ready
Testing squeezed at the end	Development late, environments not ready	Test inside each sprint; book environments and bank test slots early
Bank or vendor date missed	Dependency not tracked	Add it to the dependency board with an owner; escalate early
Status green until the week before go-live	Status reported on effort, not evidence	Report on working, tested features and exit criteria
Constant urgent requests	No scope control	Route all requests through the product owner; trade scope or date

Reporting: one weekly status for stakeholders with milestones (RAG), say/do ratio, top risks and decisions needed. Show the plan against the actual, so gaps are visible early, not at go-live.

ACH basics for the team

ACH moves money in batches between US bank accounts: 35.2 billion payments worth $93 trillion in 2025, per Nacha. Everyone on the project should know these terms before Phase 1 starts.

Participant	Role
Originator	The company or person that starts the payment (for example, an employer running payroll)
ODFI	Originating Depository Financial Institution: the bank that sends the entries into the network
ACH Operator	FedACH (Federal Reserve) or EPN (The Clearing House): receives, sorts and delivers files
RDFI	Receiving Depository Financial Institution: the bank that posts or returns the entry
Receiver	The person or company whose account is credited or debited

Participant	Role
Third-party sender / service provider	A processor or fintech acting between the Originator and the ODFI

SEC code	Used for	Authorisation
PPD	Consumer payroll credits, bill debits	Written or similar agreement
CCD	Business-to-business payments	Agreement between the businesses
WEB	Consumer debits authorised online or by mobile	Online authorisation; account validation required
TEL	Consumer debits authorised by phone	Recorded or confirmed phone authorisation
CTX	Business payments with many invoice details (addenda)	Trading partner agreement
IAT	Payments with a foreign leg	Extra OFAC and data requirements

The NACHA file: a fixed-width text file of 94-character records: File Header (1), Batch Header (5), Entry Detail (6), Addenda (7), Batch Control (8), File Control (9), padded with 9s to a multiple of 10 records.

Settlement windows (FedACH, ET):

Window	Operator deadline	Settlement
Same Day 1	10:30 a.m.	1:00 p.m.
Same Day 2	2:45 p.m.	5:00 p.m.
Same Day 3	4:45 p.m.	6:00 p.m.
Next day (future-dated)	Overnight windows	8:30 a.m. next banking day

Source: Federal Reserve FedACH Same Day FAQ and FedACH settlement tips. Same Day ACH carries up to $1,000,000 per payment. Your ODFI or sponsor bank sets earlier internal cut-offs; always design to the bank's cut-off, not the Fed's.

Returns and NOCs: an RDFI sends back an entry it can't post as a return with a reason code (for example R01 insufficient funds, R03 no account). Most returns are due within two banking days; consumer

unauthorised returns (such as R05, R07, R10) can come back up to 60 calendar days later. A Notification of Change (NOC) asks the Originator to correct account details on future entries.

Phase 1: Requirement gathering

The goal of this phase is one signed-off requirement set that covers business, operations, compliance and technology, with every rule traceable to a test.

Who to involve: product owner, business analyst, ACH operations, treasury (liquidity and funding), compliance (Nacha, Reg E, BSA/AML, OFAC), fraud, information security, architecture, QA lead, customer support, and the ODFI or sponsor bank.

Questions to answer before design

Area	Questions
Scope	Originate, receive, or both? Credits, debits, or both? Which SEC codes? Same Day ACH?
Customers	Consumers, businesses, or both? Who authorises each payment, and how is that kept?
Channels	Online banking, mobile, API, file upload (SFTP), ERP integration?
Limits	Per payment, per day, per customer; Same Day ACH limit of $1,000,000 per payment
Timing	Which bank cut-offs? Holidays and weekends? Effective entry date rules?
Funding	Prefunding or credit limits for originators? Exposure limits per customer?
Returns	Who handles returns, NOCs and reversals? What happens to the customer's balance?
Fraud	Account validation, velocity checks, first-time payee checks, monitoring rules
Reporting	Customer statuses, operations dashboards, reconciliation, regulatory reports
Data	What personal and account data is stored, where, for how long, and who can see it
Non-functional	Peak volumes, response times, availability, recovery time and recovery point

Requirement checklist

- [ ] Business requirements document (BRD) signed by the product owner and business
- [ ] Functional specs for each flow: initiate, approve, validate, batch, send, receive, post, return, NOC
- [ ] Nacha Rules mapping: each rule that applies, and the requirement that meets it
- [ ] Authorisation and record-keeping rules for each SEC code
- [ ] Fraud and account validation rules agreed with the fraud team

- [ ] OFAC screening rules (including IAT)
- [ ] Non-functional requirements with numbers, not adjectives
- [ ] Operating model: who does what in operations and support
- [ ] Traceability matrix created: requirement to design to test

Sample user stories with acceptance criteria

```
Story: Business customer sends a payroll batch
  As a business customer
  I want to upload a payroll file
  So that my employees are paid on payday

  Scenario: Valid payroll batch is accepted
    Given the customer has an approved ACH origination agreement
    And the batch total is within the customer's daily limit
    When the maker uploads the file and a checker approves it
    Then each entry is validated and the batch is queued for the next window
    And the batch uses SEC code PPD with Company Entry Description "PAYROLL"

  Scenario: Batch above the daily limit is held
    Given the batch total is above the customer's daily limit
    When the checker approves it
    Then the batch is held for treasury review
    And the customer sees the status "Pending limit review"
```

Exit criteria: signed BRD and functional specs, Nacha rule mapping reviewed by compliance, non-functional numbers agreed, traceability matrix baselined.

Phase 2: Design

Design turns the requirements into an architecture that can't send money twice, can't miss a cut-off silently, and leaves an audit trail for every step.

Core components

Component	What it does
Channels	Online, mobile, API and SFTP intake; maker-checker approval
Payment hub	Validates entries, applies limits, assigns a unique ID, tracks status
Fraud and OFAC screening	Account validation, velocity and anomaly rules, sanctions screening

Component	What it does
Funds control	Checks balance or prefunding, places holds, posts debits and credits
Batch and file builder	Groups entries by SEC code, company and effective date; builds the NACHA file
Scheduler	Releases files before each bank cut-off; handles holidays
Transmission	Sends files to the ODFI or operator over secure channels (SFTP with PGP, or API)
Inbound processor	Receives incoming entries, returns and NOCs; posts to accounts
Reconciliation	Matches sent totals to settlement advices and the general ledger
Reporting and audit	Statuses, dashboards, logs, audit evidence

Design decisions to record

- **Idempotency:** every payment has one unique key; a retry can never create a second entry.
- **Status model:** Received, Validated, Approved, Screened, Batched, Sent, Settled, Returned, Corrected (NOC), Cancelled.
- **Cut-off handling:** what happens to a payment that misses a window: next window, next day, or customer choice.
- **Trace numbers:** ODFI routing number (8 digits) + sequence (7 digits), unique per entry and stored with the payment.
- **Balancing:** batch and file control totals (entry count, entry hash, debit and credit totals) checked before release.
- **Exception queues:** rejected files, OFAC hits, fraud holds and returns each go to a named team queue.
- **Data protection:** account numbers encrypted at rest and masked in logs and screens.
- **Resilience:** recovery time and recovery point targets for each component; a manual fallback for a missed window.

Non-functional design targets (fill in your numbers)

Measure	Target
Peak entries per file window	____ entries
File build time at peak	under ____ minutes before cut-off
API response time (p95)	under ____ ms

Measure	Target
Availability	____ % per month
Recovery time / recovery point	____ minutes / ____ minutes

Exit criteria: architecture and data flow approved by architecture review, security design review passed, status model and exception handling signed off by operations.

Phase 3: Development

Build small, reviewed changes behind feature flags, with money-movement logic covered by tests before it is merged.

Engineering standards

- Branch per story; every merge needs a pull request review by a second engineer (maker-checker for code).
- Static code analysis and secret scanning run on every pull request; no merge with critical findings.
- Feature flags for new SEC codes, new windows and new customer groups.
- Configuration (cut-offs, limits, holidays, routing numbers) lives in managed config, never hard-coded.
- Money is stored as integer cents, never floating point.

Must-have build items

Item	Rule
Field validation	Routing number check digit, account number format, amount > 0, valid SEC code and transaction code
File builder	94-character records, correct record order, control totals and entry hash, padding to blocks of 10
Idempotency	Unique payment key checked before any entry is created
Duplicate file check	File hash and file ID modifier checked before transmission
Company Entry Description	"PAYROLL" for PPD payroll credits and "PURCHASE" for WEB e-commerce debits, required by Nacha from March 20, 2026 (First Horizon summary of the 2026 rules)
Status events	Every status change written as an event with time, user or system, and reason
Secure transmission	PGP-encrypted files over SFTP, or mutual-TLS API; keys in a key vault

Item	Rule
Logging	Account numbers masked; no full account or routing pair in logs

Secure coding: follow OWASP guidance; validate every input; use parameterised queries; least-privilege service accounts; no secrets in code or config files.

Exit criteria: all stories merged with reviews, static analysis clean of critical and high findings, unit tests passing at the agreed coverage, build artefacts versioned and signed.

Phase 4: Unit testing

Unit tests prove each rule in isolation, fast, on every build; any code that touches amounts, dates or file format needs them.

Area	Test cases
Routing number	Valid check digit passes; wrong check digit, 8 or 10 digits, letters fail
Amounts	Zero, negative, max allowed, one cent over limit, Same Day limit of $1,000,000
Dates	Effective date on a weekend or holiday rolls to next banking day; past dates handled as the rules say
SEC codes	Each code produces the right transaction codes and required fields
File builder	Record length 94; record order; batch and file control totals; entry hash; block padding
Idempotency	Same key twice creates one entry
Status model	Only allowed status changes succeed; others are rejected and logged
Masking	Logs never contain a full account number

Coverage gate: agree a minimum (for example 80% of lines and 100% of money-movement rules), enforced in the build pipeline.

Exit criteria: all unit tests pass in CI, coverage gate met, results stored with the build.

Phase 5: Performance testing

Performance testing proves the system builds and sends the busiest day's files before the bank's cut-off, with room to spare.

Test type	What it proves
Load test	Normal peak volume (for example payday or month-end) runs within targets
Stress test	Where the system breaks, at 2 to 3 times peak
Spike test	A sudden burst (a large customer uploading a big file) doesn't block other customers
Soak test	8 to 24 hours at steady load with no memory or connection leaks
Window test	File build and transmission finish before each cut-off at peak volume
Failover test	A node or database failover during a window doesn't lose or duplicate entries

How to run it

1. Use production-like volumes and data shapes, with masked or synthetic data.
2. Test in an environment sized like production.
3. Measure API response time (p95, p99), file build time, transmission time, CPU, memory and queue depth.
4. Repeat after every fix; keep the reports as evidence.

Exit criteria: every non-functional target met at peak, with the gap to cut-off documented; no open high-severity performance defects.

Phase 6: End-to-end testing

End-to-end testing proves a payment works across every system, from the customer's screen to the receiving bank and back, including what happens when things go wrong.

Environments: a full test chain with the ODFI or sponsor bank's test environment, an operator or simulator, test RDFI accounts, core banking, general ledger and reporting.

Scenario group	Examples
Happy paths	PPD payroll credit, CCD vendor payment, WEB debit, Same Day credit in each window
Approvals	Maker-checker approval; rejection; approval after cut-off
Limits	Per payment, daily and customer limits; Same Day limit
Screening	OFAC match held; false positive released; IAT entry screened
Fraud	First-time payee, unusual amount, velocity breach held for review
Timing	Payment just before and just after each cut-off; weekend and holiday dates

Scenario group	Examples
Returns	R01, R02, R03, R04, R10 and R29 returned and posted correctly; customer told
NOCs	C01 and C02 received; account details updated for future entries
Reversals	Duplicate or wrong-amount file reversed within the allowed time
Reconciliation	File totals match settlement advices and the general ledger
Recovery	File rejected by the operator and resent; transmission failure and retry with no duplicates

E2E entry and exit criteria

Entry criteria: unit and performance exit criteria met; E2E environment connected to the bank's test system; test data and accounts ready; test plan approved.

Exit criterion	Target
Requirements covered by a passed E2E test	100% (traceability matrix)
Test cases executed	100% of planned cases
Test cases passed	At least 98%; every failure has an approved defect and plan
Open defects	0 critical, 0 high; medium only with an approved workaround and fix date
Money-movement scenarios (credits, debits, returns, NOCs, reversals)	100% passed, no exceptions
Reconciliation	File totals = settlement advices = general ledger for the whole cycle
Regression	Full ACH regression suite passed on the final build
Bank certification tests	Passed, with the bank's written confirmation
Evidence	Test results, defect log and reconciliation reports stored in the test repository

E2E sign-off: QA lead (owner), engineering lead and ACH operations lead sign the E2E test summary report before UAT starts.

Phase 7: UAT

User acceptance testing is where the business confirms the product does what it needs, in the way people will really use it.

Who tests: business users, ACH operations, customer support, treasury and a pilot group of real business customers (in a test environment).

Entry criteria: E2E exit criteria met; UAT environment stable; test accounts, users and roles set up; training given; defect process agreed.

UAT focus	What users check
Customer journeys	Create, approve, cancel and track payments; download reports and statuses
Operations	Work exception queues: OFAC holds, fraud holds, rejected files, returns, NOCs
Treasury	Funding, limits, settlement reports, cash positioning
Support	Search a payment by trace number, explain a status, raise an incident
Notifications	Customer emails and internal alerts arrive with the right words and timing
Reports	Daily reconciliation and management reports are correct

UAT exit criteria

Exit criterion	Target
UAT scripts executed	100% of agreed business scenarios
UAT scripts passed	100% of critical journeys; others passed or accepted with a documented workaround
Open defects	0 critical, 0 high; medium and low listed with owners and dates
Operations readiness	Exception queues, runbooks and reports proven by the operations team
Support readiness	L1 scripts tested; L2 and L3 on-call rotas confirmed
Customer communications	Notification emails and help content approved
Compliance review	Nacha, Reg E, OFAC and data controls confirmed by compliance

Sign-off process (E2E and UAT)

Nothing moves to release without written sign-off from every required person; a missing signature is a "no-go".

Step	What happens	Who
1. Test summary report	Results, coverage, open defects, risks and evidence links in one report	QA lead (E2E) / UAT lead (UAT)
2. Defect triage	Every open defect agreed as fix-before-release or accepted with a workaround	Product owner + QA lead + engineering lead
3. Review meeting	Report walked through; questions answered; risks recorded	All sign-off owners
4. Sign-off	Each owner signs (e-signature or approval in the test tool)	See sign-off matrix
5. Store	Signed report attached to the release record for CAB and audit	Release manager

Sign-off matrix

Stage	Must sign	Consulted
E2E testing	QA lead, engineering lead, ACH operations lead	Product owner, security
UAT	Product owner, business owner, ACH operations lead, compliance	Customer support lead, treasury
Release (CAB)	Release manager, change manager, product owner	Security, operations

Waivers: if an exit criterion is not met, the product owner may ask for a written waiver that states the risk, the workaround, the fix date and the owner. Waivers for money-movement or compliance criteria also need the compliance lead's approval. Waivers are listed in the CAB pack.

Phase 8: Release management

However the team works day to day, every change reaches production through one controlled path: a planned release, an approved change and a clear go/no-go, with a hotfix path for urgent fixes.

Release calendar rules

- Release on a fixed train (for example every second Tuesday night), outside ACH windows.

- Blackout days: payroll peaks, month-end, quarter-end, the day before and after federal holidays, and Nacha rule effective dates.
- No release within 2 hours before or after a same day cut-off.

Change approval (CAB) pack

- ☐ Release notes: what changes, for whom, and any customer action needed
- ☐ Test evidence: unit, performance, E2E and UAT sign-offs
- ☐ Security sign-off: scan results, OSS report, penetration test if scope changed
- ☐ Deployment plan with owners and times
- ☐ Rollback plan, tested in a lower environment
- ☐ Communication plan: customers, operations, support, sponsor bank
- ☐ Version tag and signed build artefact

Go/no-go meeting (T-1 day): release manager runs it; product owner, engineering lead, QA lead, operations, support lead and on-call engineer attend. Go only if every item above is green.

Rollback rule: decide the rollback trigger before the release (for example a failed smoke test, file rejected by the bank, or error rate above an agreed level). Never roll back a file that has already been sent; correct it with a reversal or a new entry instead.

Hotfix process

Step	What happens	Who
1. Declare	Incident is Severity 1 or 2 and can't wait for the next release	Incident manager
2. Approve	Emergency change approved by the change manager and product owner	Change manager
3. Fix	Branch from the live release tag; smallest possible change	L3 engineer
4. Test	Targeted unit and regression tests on the fix; peer review	QA + second engineer
5. Deploy	Deploy outside a cut-off; smoke test; watch the next window	Release manager + L2
6. Merge back	Merge the fix into the main branch so the next release keeps it	L3 engineer
7. Review	Emergency change reviewed at the next CAB; root cause logged	Change manager

Exit criteria: CAB approval recorded, go decision minuted, deployment window booked.

Phase 9: Deployment and cutover

Deploy in a quiet window, prove it with real but tiny payments, then watch closely through the first live windows.

Deployment runbook (example times, ET)

Time	Step	Owner
T-60 min	Bridge call opens; confirm no files in flight; freeze new submissions if needed	Release manager
T-45 min	Back up databases and config; confirm rollback package ready	DBA + DevOps
T-30 min	Deploy through the pipeline (no manual changes); run database migrations	DevOps
T-15 min	Smoke tests: login, create, approve, validate, build a test file (not sent)	QA
T-0	Turn on feature flags for a pilot group	Release manager
T+30 min	Penny test: company-owned accounts, $0.01 to $1.00 credit and debit through a real window	Operations
Next window	Confirm file accepted by the bank, settlement advice matches, entries posted	Operations + L2
End of day	Reconcile totals to the general ledger; go/no-go to widen the rollout	Product owner

Penny testing in production

A penny test sends tiny real ACH entries through the live path right after a deployment, because only production proves that the bank, the operator and the receiving accounts still work with your release. No release is opened to customers until its penny tests pass.

Penny test matrix (example; use your own SEC codes and windows)

#	Entry	SEC code	Window	Amount	What it proves
1	Credit to company-owned account at another bank	PPD	Same Day 1 or 2	$0.01	File built, sent, accepted and settled; credit posts
2	Debit from company-owned account at another bank	CCD	Same Day 1 or 2	$0.01	Debit path, settlement advice, GL posting

#	Entry	SEC code	Window	Amount	What it proves
3	Payroll-style credit	PPD with "PAYROLL"	Next day	$0.01	Company Entry Description and future-dated path
4	Incoming entry to your own account	Any	Next window	$0.01	Inbound file processing and posting (as RDFI)
5	Reversal of test 1 or 2	Same as original	Next window	$0.01	Reversal logic and audit trail

How to run it

1. Before the release, compliance and finance approve the penny test plan once; reuse it every release.
2. Use only company-owned accounts at two or more banks, set up with signed authorisations. Never use customer money.
3. After the smoke tests, turn on the change for a pilot group and submit the penny entries in the first available window.
4. Check each entry end to end: file acknowledgement, settlement advice, posting at both ends, status updates, customer-style notification, and the GL.
5. Reverse or offset every test amount, and label every entry (for example Company Entry Description "PENNYTEST" and the release number in the addenda or internal reference) so finance can reconcile it.
6. A failed penny test stops the rollout: keep the feature flag off, open an incident, and decide on fix or rollback before the next window.

Guardrails

- Fund tests from a separate account with a small daily limit.
- Keep penny entries out of business dashboards, return-rate metrics and customer reports.
- Store the results (screenshots, file IDs, trace numbers, settlement advices) as release evidence.

Penny test vs micro-entries: a penny test checks *your release*, using your own accounts. Micro-entries (credits under $1, with the Company Entry Description "ACCTVERIFY") check *a customer's account* before you debit it, and follow their own Nacha rules. Don't mix the two.

Hypercare (first 2 weeks): daily check-in; L3 on standby during every window; extra monitoring on rejects, returns and response times; daily report to the product owner.

Exit criteria: penny tests passed, first full-volume window clean, reconciliation balanced, hypercare exit signed by operations.

Compliance

Compliance is built into every phase, and the team keeps the evidence that proves it.

Area	What applies	Evidence to keep
Nacha Rules	Origination agreements, authorisation per SEC code, return and NOC handling, annual Rules compliance audit	Signed agreements, authorisation records, audit report
ACH fraud monitoring	Risk-based fraud monitoring of entries: Phase 1 (ODFIs) from March 20, 2026, Phase 2 (other originators, third parties and RDFIs) from June 22, 2026 (Bottomline)	Written procedures, annual review, alert logs
Company Entry Descriptions	"PAYROLL" and "PURCHASE" where they apply, from March 20, 2026	File samples and tests
WEB debits	Account validation before first use of an account number	Validation logs
Data security	Account numbers protected and unreadable when stored electronically (Nacha data security rules)	Encryption design, key management records
Regulation E	Consumer error resolution and unauthorised transfer handling	Case logs and timelines
BSA/AML	Customer due diligence, transaction monitoring, suspicious activity reports	Monitoring rules, SAR records (restricted)
OFAC	Screen parties before sending; extra checks for IAT entries	Screening logs, hit decisions
Record keeping	Keep authorisations and entry records for the required periods	Retention schedule
Third-party risk	Due diligence and contracts with processors and vendors	Vendor files, reviews

Audit-ready evidence pack: traceability matrix, test results, approvals, CAB records, access reviews, incident reports and the annual Nacha Rules audit.

Watch next: Nacha's new definition of IAT entries took effect September 18, 2026; confirm IAT flags and screening rules match it.

Internal and external audit

Treat audit as a normal part of running ACH: keep evidence as you go, so an audit is a matter of handing it over, not rebuilding it.

Who audits ACH

Audit	Who	How often	Focus
Internal audit	Your third line (internal audit team)	Risk-based, usually yearly for payments	Controls design and operation, policies followed
Nacha Rules compliance audit	Internal audit or an external firm	Every year, completed by December 31; keep proof for 6 years	Compliance with the Nacha Operating Rules
External audit (financial and IT controls)	External auditors	Yearly	IT general controls, change management, access, reconciliation (SOX where it applies)
SOC 1 / SOC 2	External auditors	Yearly	Needed if you process payments for other companies
Regulatory exam	OCC, FDIC, Federal Reserve, state regulators, CFPB	Exam cycle	IT, BSA/AML, OFAC, consumer protection
Sponsor bank / ODFI review	Your bank partner	Per agreement, often yearly	Third-party sender duties, risk limits, Nacha compliance

Internal audit checklist

- [ ] Policies and procedures for ACH origination, receiving, returns and exceptions are current and approved
- [ ] Origination agreements signed for every originator and third-party sender
- [ ] Customer and originator risk assessments done; exposure limits set and reviewed
- [ ] Authorisation records kept and retrievable for each SEC code sampled

- [ ] Maker-checker enforced by the system; no self-approval found in samples
- [ ] Access reviews done quarterly; leavers removed on time; privileged access logged
- [ ] Change management: every production change has approval, testing and a rollback plan
- [ ] Returns and NOCs worked within Nacha timeframes; return rates under thresholds
- [ ] Fraud monitoring procedures written, reviewed yearly, and alerts worked
- [ ] OFAC screening logs complete; hits decided and recorded
- [ ] Daily reconciliation balanced; breaks investigated and closed
- [ ] Account numbers protected at rest; data retention followed
- [ ] DR plan tested in the last 12 months with evidence
- [ ] Vendor and processor reviews current, including SOC reports
- [ ] Previous audit findings closed with evidence

External audit and exam checklist

- [ ] Evidence index ready: one list linking each control to its evidence location
- [ ] Last annual Nacha Rules compliance audit report and management response
- [ ] Population lists ready to sample: payments, changes, access changes, incidents, returns
- [ ] System and data flow diagrams, and the list of systems in scope
- [ ] Change records with CAB approvals, test sign-offs (E2E, UAT) and penny test evidence
- [ ] User access listings, review sign-offs and privileged access logs
- [ ] Incident and problem records, with post-incident reviews for Sev 1 and Sev 2
- [ ] BSA/AML and OFAC policies, training records and independent testing results
- [ ] Reg E error-resolution cases and timeliness
- [ ] Vendor contracts, due diligence and SOC 1 / SOC 2 reports with gap reviews
- [ ] DR and business continuity test results
- [ ] Open findings tracker with owners, due dates and status

During and after an audit

1. Name one audit coordinator; all requests go through them and are logged.

2. Answer each request within an agreed time (for example 3 business days); share evidence, not opinions.
3. Agree the facts of each draft finding before the report is final.
4. For each finding: root cause, action plan, owner and due date in the findings tracker.
5. Close a finding only with evidence that the fix works; internal audit validates closure.
6. Report open findings monthly to management until all are closed.

Worked example: operations updated a payment status directly in the database

What happened: to clear a stuck batch, an operations analyst ran `UPDATE payments SET status = 'SETTLED'` in production, outside the application, with no ticket or approval.

Why auditors flag it (usually a high-rated finding)

- It bypasses the application's controls: maker-checker, status rules and the audit trail.
- It breaks segregation of duties: the same person decided, changed and "approved" the record.
- The system of record no longer matches reality: the customer, the ledger and the bank may disagree, and money can be sent twice or never.
- It's an IT general controls failure (change and access management), which can affect SOX and regulatory opinions.

Immediate response

1. Log an incident; record who ran what, when, and on which records (from database audit logs).
2. Reconcile the affected payments with the bank file acknowledgements, settlement advices and the ledger; correct any money impact through the proper process.
3. Restore the correct status through an approved data fix (below), keeping the before and after values.
4. Tell compliance and internal audit; check whether customers or regulatory reports were affected.
5. Run a root cause review: why was there no proper tool, and why did the person have write access?

Controls that stop it happening again

Control	What it means
No direct write access	Operations and developers have read-only access to production data; write access only for DBAs through a privileged access (PAM) tool, time-limited and recorded
Status changes only in the app	An admin screen or API to change a status, with a reason code, maker-checker and an audit event

Control	What it means
Approved data-fix process	Ticket + business reason, script reviewed by a second engineer, tested in a lower environment, approved by the change manager and product owner, run by a DBA, before/after snapshot kept
Database audit logging	Every insert, update and delete on payment tables logged to a tamper-proof store
Monthly review	Compare database change logs with approved tickets; any change without a ticket is an incident
Alerts	Real-time alert to security and the operations lead on any manual write to payment tables

Add to the internal audit checklist:

- ☐ No direct production data changes without an approved data-fix ticket (sample the database audit logs against tickets)
- ☐ Write access to payment tables limited to DBAs through PAM, reviewed quarterly

Open source software (OSS)

Every open source library in the ACH stack must be known, licensed correctly and kept free of serious known vulnerabilities.

Control	Rule
Approved sources	Libraries only from approved repositories and mirrors
Licence check	Allowed: permissive licences (for example MIT, Apache 2.0, BSD). Needs legal review: copyleft (for example GPL, AGPL)
Vulnerability scan	Software composition analysis (SCA) on every build; no release with critical or high vulnerabilities unless risk-accepted
SBOM	A software bill of materials produced for every release and stored with it
Patching	Critical vulnerabilities fixed within ____ days, high within ____ days (set by your security policy)
Pinning	Exact versions pinned; updates through pull requests with tests
End of life	No end-of-life frameworks or runtimes in production

Control	Rule
Ownership	Each library has an owning team that tracks its updates

Exit criteria for every release: SCA clean or risk-accepted, licence report approved, SBOM attached to the release record.

Password and access policy

Access to ACH systems follows NIST SP 800-63B Revision 4 (final, July 2025): long passwords, a breached-password blocklist, MFA everywhere, and no forced periodic changes (NIST SP 800-63B-4).

Passwords (people)

Rule	Setting
Minimum length	15 characters when a password is the only factor; at least 8 when used with MFA (we recommend 12+)
Maximum length	Allow at least 64 characters; spaces and all printable characters allowed
Composition rules	None (no forced mix of upper case, numbers and symbols)
Blocklist	Reject common, breached and company-specific passwords, and say why
Expiry	No scheduled expiry; force a change only on evidence of compromise
Hints and security questions	Not allowed
Password managers	Approved manager for all staff; paste allowed
Lockout	Throttle repeated failed attempts; alert security on attack patterns

Access controls

- **MFA for everyone** on every ACH system; phishing-resistant MFA (FIDO2 security keys or passkeys) for admins and anyone who can approve or release files.
- **Least privilege and roles:** separate roles for maker, checker, operations, support and admin; no one can create and approve the same payment.
- **Privileged access:** production access only through a privileged access management (PAM) tool, time-

limited and recorded.

- **Service accounts:** no interactive login; secrets in a vault; rotated on schedule and on any staff change.
- **Keys:** PGP and TLS keys in a key vault or HSM; rotated on a schedule; separate keys per environment.
- **Joiners, movers, leavers:** access removed the same day someone leaves.
- **Access reviews:** quarterly for all ACH systems; monthly for privileged access.
- **Shared accounts:** not allowed.

Server and certificate management

Every server, certificate and key in the ACH chain is inventoried, hardened and renewed on time, because an expired certificate stops files as surely as a code bug.

Certificates and keys

Item	Used for	Rule
TLS server certificates	APIs, portals, SFTP endpoints	Issued by an approved CA; automated renewal; alert at 30, 14 and 7 days before expiry
Mutual TLS (client) certificates	API links to the bank or processor	Exchanged with the bank under change control; renewal tested in the bank's test environment first
PGP keys	Encrypting and signing NACHA files	Separate keys per environment; public keys swapped with the bank in advance; old key kept until all files are confirmed
SSH keys	SFTP connections	Key-based login only; rotated on schedule and on staff change
Code-signing keys	Signing build artefacts	In an HSM or key vault; access logged

Shorter certificate lifetimes: public TLS certificates are moving to much shorter maximum lifetimes under CA/Browser Forum rules (200 days from March 2026, falling further in 2027 and 2029), so manual renewal is no longer safe. Automate issuance and renewal (for example with ACME), and keep a certificate inventory with owners and expiry dates.

Certificate renewal runbook

1. Inventory flags a certificate or key due within 30 days.
2. Owner raises a standard change; schedule outside ACH windows.

3. For bank-facing certificates and PGP keys: send the new public part to the bank, agree a switch date and test in their test environment.
4. Install, restart in a controlled order, and run a connectivity test and a test file.
5. Confirm the first live file after the switch is accepted; then retire the old certificate or key.

Server hardening and certification

- Build servers from hardened images (for example CIS Benchmarks); no default accounts; only needed ports open.
- Patch operating systems and middleware on a set schedule; critical patches fast-tracked.
- Vulnerability scan every server monthly and after every change; penetration test yearly and after major changes.
- Log all access and admin actions to a central, tamper-proof log.
- **Bank connectivity certification:** before go-live, and after any change to IP addresses, certificates, keys or file formats, complete the ODFI or processor's certification tests (connectivity, file format, acknowledgements, returns) and keep their written sign-off.
- Firewall rules and IP allow-lists with the bank kept in a register and reviewed every quarter.

Software certification and licences

No software reaches or stays in the ACH chain without three proofs: it passed our own release certification, the bank or vendor certified it where required, and its licence and support are current.

1. Release certification (our software)

A release is certified for production only when QA, security and operations sign it off:

- [ ] All tests in the traceability matrix passed (unit, performance, E2E, UAT)
- [ ] Security scans clean: static analysis, SCA (open source), container and server scans
- [ ] Signed build artefact with version tag and SBOM
- [ ] Operations readiness: runbooks updated, alerts configured, support trained
- [ ] Certification record stored with the CAB approval

2. External certifications

What	When it's needed	Evidence
Bank / processor certification	New connection, new file format or SEC code, new API version, changed keys or IPs	Bank's test results and written sign-off
Vendor assurance reports	Every vendor that stores, processes or sends ACH data	SOC 1 / SOC 2 Type II reports reviewed every year; bridge letters for gaps
Security certifications	Hosting and critical vendors	ISO 27001 certificate or equivalent, checked for scope and expiry
Card data (only if cards are in scope too)	Same platform handles card payments	PCI DSS attestation of compliance

3. Licences and support renewals

Item	Rule
Licence register	Every commercial product: owner, licence type, seats or capacity, renewal date, cost
Renewal alerts	90, 60 and 30 days before expiry, to the owner and procurement
Support status	Only vendor-supported versions in production; plan upgrades before end of support
Capacity	Licence limits checked against volume growth every quarter
Audit	Licence register reviewed every year with procurement and finance

Tip: keep the certificate, licence and vendor-certification registers in one tracker with owners and expiry dates; it's the same discipline that keeps certificates from expiring.

Production support model: L1, L2, L3

Support works in three levels, each with a clear job, a time limit and a named next step, so no ACH problem waits past a cut-off.

Level	Who	Does	Escalates when
L1	Service desk / customer support	Takes calls and tickets; checks payment status by trace number; follows known scripts; resets access; logs every case	Not solved by a script in 15 minutes, or any sign of a system problem, money at risk or fraud

Level	Who	Does	Escalates when
L2	ACH operations + application support	Works exception queues (rejects, returns, NOCs, OFAC and fraud holds); checks logs and file status; restarts jobs; reprocesses or resends files under the runbook	Needs a code or data fix, the cause is unknown after 30 minutes, or a cut-off is at risk
L3	Engineering (developers, DBAs, platform)	Root cause; code or data fixes; hotfixes; vendor and bank technical contact	Severity 1, or a fix needs an emergency change

Severity levels and response times (example; set your own)

Severity	Example	Response	Update cadence	Target fix or workaround
Sev 1	No files can be sent; duplicate file sent; data breach	15 minutes, 24/7	Every 30 minutes	Before the next cut-off
Sev 2	One window missed; large customer blocked; return processing down	30 minutes	Every hour	Same day
Sev 3	One customer or a small group affected; workaround exists	4 business hours	Daily	5 business days
Sev 4	Question, cosmetic issue, report request	1 business day	On change	Next release

Escalation path: L1 → L2 → L3 → incident manager → head of payments operations → CIO and chief compliance officer for Sev 1. Compliance and fraud join at once for any fraud, OFAC or data incident. The sponsor bank or ODFI is told within the time their agreement sets.

On-call: L2 and L3 on-call rotas cover every processing day and all ACH windows, with a primary and a backup; on-call schedules are kept in the paging tool.

After every Sev 1 or Sev 2: a blameless post-incident review within 5 business days, with root cause, customer impact, money impact and actions with owners and dates.

Operational runbooks

Each common ACH problem has a short, tested runbook: how you spot it, what to do first, who owns it and how to close it.

Problem	How you spot it	First actions	Owner	Close when
File rejected by bank or operator	Reject acknowledgement or no acknowledgement in 30 minutes	Read the reject reason; fix format or totals; rebuild and resend before cut-off; tell the bank	L2, then L3	Accepted acknowledgement received
Missed cut-off	File not sent by the internal deadline	Send in the next window; tell affected customers; Same Day items roll to the next window or day per policy	L2 + operations lead	File settled; customers informed
Duplicate file sent	Same file hash or totals sent twice	Sev 1. Call the ODFI at once; request deletion if not yet processed; otherwise send a reversal within the allowed time	L2 + L3 + incident manager	Reversal settled; customers made whole
Wrong amount or wrong account	Customer or reconciliation finds it	Reversing entry within the Nacha timeframe and for an allowed reason; notify the Receiver as the Rules require	Operations	Reversal settled; root cause logged
Spike in returns	Return rate above threshold for a customer or SEC code	Hold new files from that originator; review authorisations and account validation; tell compliance and fraud	Operations + fraud	Rate back under threshold
NOC received	C-code NOC in the inbound file	Update account details within the Nacha timeframe; confirm with the originator	Operations	Details updated
OFAC hit	Screening alert	Hold the entry; compliance decides; block or release; record the decision	Compliance	Decision recorded

Problem	How you spot it	First actions	Owner	Close when
Suspected fraud	Fraud rule alert or customer report	Hold related entries; contact the customer by a known channel; request return; file reports as required	Fraud team	Case closed
Bank or operator outage	Transmission fails; operator notice	Retry under runbook; switch to backup channel if contracted; customer notice	L2 + L3	Files sent and settled
Reconciliation break	Settlement advice or GL doesn't match file totals	Find the gap by batch and entry; correct postings; report to finance	Operations + finance	Balanced

For every runbook: log the incident ticket, record each action with a time, and attach the evidence.

Disaster recovery (DR) and business continuity

If the primary site fails, ACH files must still reach the bank before the cut-off, with no payment lost and none sent twice. DR is designed in Phase 2, tested before go-live and re-tested every year.

Recovery targets (example; set yours with the business)

Component	Recovery time (RTO)	Data loss allowed (RPO)	DR approach
Payment hub and database	1 hour	0 to 5 minutes	Replicated to a second region; warm standby
File builder and scheduler	1 hour	0 (state in the database)	Standby in the second region
Bank transmission (SFTP / API)	30 minutes	0	Second connection from the DR site, already certified by the bank
Inbound processing (returns, NOCs)	4 hours	0 (files can be re-pulled)	Re-download files from the bank after failover
Customer channels (online, API)	2 hours	5 minutes	Load balancer switch to the second region

Component	Recovery time (RTO)	Data loss allowed (RPO)	DR approach
Reporting and dashboards	24 hours	1 hour	Rebuild from replicated data

ACH-specific DR rules

- **Pre-certify the DR path with the bank:** DR IP addresses on the bank's allow-list, certificates and PGP keys already exchanged, and a test file sent from DR at least once a year.
- **No duplicates after failover:** before releasing any file from DR, check which files the bank already acknowledged (file ID, file hash and control totals) and send only the missing ones.
- **In-flight payments:** payments in "Batched" or "Sent" status at failover are frozen until reconciled against bank acknowledgements.
- **Cut-off fallback:** if systems can't recover before the cut-off, operations upload the last good file through the bank's backup channel (for example the bank's portal), under maker-checker.
- **Keys and secrets** are replicated to the DR key vault; break-glass access is documented and logged.
- **Third parties:** confirm the bank, processor and cloud providers have DR plans that meet your targets; keep their evidence.

Failover runbook

1. **Declare:** incident manager declares a disaster (Sev 1) and opens the bridge; notify the sponsor bank or ODFI.
2. **Freeze:** stop new file releases; capture the last known status of every batch and file.
3. **Fail over:** switch database, application and transmission to the DR site using the scripted runbook.
4. **Verify:** smoke tests, then reconcile in-flight files with the bank's acknowledgements.
5. **Resume:** release only missing or new files; watch the next window closely.
6. **Communicate:** updates in the incident channel every 30 minutes; customer notice if any payment is delayed.
7. **Fail back:** after the primary site is fixed, fail back in a quiet window with the same checks, then hold a post-incident review.

DR testing

Test	How often	What it proves
Backup restore test	Monthly	Backups can actually be restored

Test	How often	What it proves
Component failover	Quarterly	Database, app and transmission each fail over
Full DR exercise with the bank	Yearly, and after major changes	A real test file goes from DR to the bank and back
Tabletop exercise	Twice a year	People know their roles and the runbook works

Exit criteria for DR readiness: targets agreed and approved, DR path certified by the bank, full DR test passed with evidence, and results reported to management and, where required, to the sponsor bank and regulators.

Notifications and alerts

Customers hear about their payments by email and in-app messages; the team hears about problems through Slack or Teams alerts that page the right person and climb the escalation path automatically.

Customer notification emails

Event	Who gets it	Must include
Payment or batch submitted	Maker and approver	Reference, amount, effective date
Approval needed	Checker	What to approve, by when (bank cut-off)
Payment sent	Originator	Reference, trace number, expected settlement date
Payment held (limit, fraud, OFAC)	Originator	Plain-language reason, what to do next (never the exact fraud or OFAC rule)
Payment returned	Originator	Return reason in plain words, what to do next
Notification of Change	Originator	Which details changed and that future payments use them
Missed window or delay	Affected customers	New expected date, apology, contact

Email rules: never show full account numbers (last 4 digits only); no links that ask for passwords; send from a verified domain with SPF, DKIM and DMARC; keep templates under change control.

Internal alert workflow (Slack or Teams)

Channel	Used for
#ach-alerts	Automated alerts from monitoring: rejects, delays, return spikes, failed jobs
#ach-incident-[number]	One channel per Sev 1 or Sev 2 incident, opened by the incident bot
#ach-release	Release announcements, go/no-go results, deployment status
#ach-ops-daily	Daily window and reconciliation summary

How an alert flows

1. Monitoring detects the event (for example "File not acknowledged 30 minutes before cut-off").
2. The alert posts to #ach-alerts with severity, window, impact, runbook link and owner.
3. The paging tool pages the L2 on-call; no acknowledgement in 10 minutes pages the backup, then the L3 lead.
4. The on-call acknowledges in the channel (an emoji reaction or button) and starts the runbook.
5. If it is Sev 1 or 2, the bot opens an incident channel, starts a bridge call and invites the incident manager.
6. Updates post on the severity's cadence until the alert is resolved.
7. The channel closes with a summary and a link to the post-incident review.

Alert message template

```
🔴 SEV 2 | ACH file not acknowledged
Window: Same Day 2 (bank cut-off 2:15 p.m. ET)
Impact: 1,240 entries, $3.2M, 18 customers
Runbook: <link>
Owner: @l2-oncall   Next update: 1:45 p.m. ET
```

Escalation by time

Time after alert	No acknowledgement or no fix: who is pulled in
0 min	L2 on-call paged
10 min	Backup L2 and L3 lead paged
30 min	Incident manager; operations lead

Time after alert	No acknowledgement or no fix: who is pulled in
60 min or cut-off at risk	Head of payments operations; sponsor bank contact
Sev 1 confirmed	CIO, chief compliance officer, communications lead

Keep alerts useful: every alert has an owner and a runbook; review noisy alerts monthly and remove any that nobody acts on.

Monitoring, KPIs and reporting

Monitor business results, not just servers: a healthy server can still send a late or wrong file.

KPI	Why it matters	Example alert
Files sent on time per window	Late files mean late pay	File not sent 30 minutes before cut-off
File acknowledgement time	Proves the bank accepted the file	No acknowledgement 30 minutes after send
Reject rate (files and entries)	Format or data problems	Any file reject
Return rate by originator and SEC code	Fraud or poor authorisations	Above your agreed threshold for an originator
Unauthorised return rate	Nacha watches this closely	Approaching the Nacha threshold
NOCs not actioned	Rules require timely updates	NOC older than the allowed time
OFAC and fraud holds open	Customer delay and risk	Hold older than 2 hours on a processing day
Reconciliation breaks	Money not where it should be	Any unmatched amount at day end
API p95 response time and error rate	Customer experience	Above target for 10 minutes
Certificates and keys expiring	Silent outage risk	30 days before expiry

Reports

- **Daily:** window summary, rejects, returns, holds, reconciliation status (posted to #ach-ops-daily).
- **Weekly:** incidents, open defects, top return reasons, alert noise.
- **Monthly:** service levels, return rates against Nacha thresholds, access review status, OSS and vulnerability status, capacity trend.
- **Yearly:** Nacha Rules compliance audit, risk assessment, disaster recovery test results.

Checklists and RACI

Use these as the standard sign-off tools; copy them into your tracker for each release.

RACI (R = does it, A = owns the result, C = consulted, I = informed)

Activity	Product owner	BA	Architect	Dev	QA	Release mgr	Ops / L2	L3	Compliance	Security
Requirements	A	R	C	C	C	I	C	I	C	C
Design	C	C	A/R	C	C	I	C	C	C	C
Development	I	C	C	A/R	C	I	I	C	I	C
Testing (unit to E2E)	I	C	I	R	A/R	I	C	C	I	C
UAT	A	R	I	C	R	I	R	I	C	I
Release and deployment	C	I	C	R	C	A/R	R	R	I	C
Production support	I	I	I	C	I	I	A/R	R	C	C
Compliance evidence	C	R	I	I	R	R	R	I	A	C
Certificates, keys, licences	I	I	C	C	I	C	R	R	I	A

Daily operations checklist (ACH operations)

- [ ] All windows' files sent and acknowledged
- [ ] Settlement advices match file totals

- [ ] Returns and NOCs worked within Rules timeframes
- [ ] OFAC and fraud holds cleared or escalated
- [ ] Reconciliation balanced and posted to #ach-ops-daily
- [ ] Any alert or incident ticket updated

Go-live readiness checklist

- [ ] Phase exit criteria met for phases 1 to 8
- [ ] Bank certification sign-off received
- [ ] Certificates, PGP keys and IP allow-lists exchanged and tested
- [ ] Runbooks, alerts and on-call rotas live
- [ ] Support teams trained; scripts published for L1
- [ ] Customer communications approved and scheduled
- [ ] Rollback plan tested; penny-test accounts ready; DR path certified by the bank and DR test passed

Appendix: return codes and glossary

Common return codes

Code	Meaning	Usual action
R01	Insufficient funds	Retry only as the Rules allow; tell the originator
R02	Account closed	Stop future entries; ask for new details
R03	No account / unable to locate	Check details; stop future entries
R04	Invalid account number	Fix validation; ask for correct details
R05	Unauthorised debit to consumer account (corporate SEC code)	Review authorisation; check SEC code use
R07	Authorisation revoked by customer	Stop future debits
R10	Customer says not authorised	Review authorisation; fraud check
R16	Account frozen	Stop entries; check for legal or OFAC reasons
R29	Corporate customer says not authorised	Review agreement with the business

Common NOC codes: C01 incorrect account number; C02 incorrect routing number; C03 incorrect routing and account number; C05 incorrect transaction code (checking vs savings).

Glossary

Term	Meaning
CAB	Change Advisory Board: approves changes to production
Entry	One payment inside an ACH file
Effective entry date	The date the originator wants the entry to settle
Hypercare	Extra support right after go-live
NOC	Notification of Change: request to correct account details
ODFI / RDFI	Sending bank / receiving bank
Penny test	A tiny real payment in production to prove the live path works
SBOM	Software bill of materials: list of all components in a release
SEC code	Standard Entry Class code: the type of ACH payment
Trace number	Unique number that identifies an entry across banks

Sources

- Nacha: 2025 ACH Network volume
- Federal Reserve: Same Day ACH FAQ
- Federal Reserve: FedACH settlement tips
- First Horizon: 2026 Nacha rule changes
- Bottomline: 2026 Nacha compliance dates
- NIST SP 800-63B-4: authenticators and passwords

Always confirm current rules in the latest Nacha Operating Rules and with your ODFI or sponsor bank.