

MetaFlow ACCESS Data Infrastructure

Technical Brief and Phase 1 Operating Manual Draft

Version 2.0 - For Nathan Ogden / Arc.dev Technical Onboarding

Field	Value
Prepared for	MetaFlow Health Inc.
Prepared by	Ling Zhou
Status	Draft for architecture review and synthetic-data-only development
Primary audience	Nathan Ogden, technical contractor from Arc.dev
Primary goal	Create an ACCESS-specific operational database foundation, not a full EHR
Development boundary	Phase 1 must use synthetic/sample data only. No real patient data, PHI, MBI, CMS credentials, production credentials, or secrets should be used, stored in GitHub, or placed in any local development environment.

This document is an operating brief for architecture and contractor onboarding. It is not legal advice, a final compliance policy, a clinical protocol, a CMS API specification, or a production SOP.

Contents

- 1. Purpose and scope
- 2. Executive summary
- 3. Phase 1 development boundary
- 4. Design doctrine
- 5. System map and source-of-truth model
- 6. ACCESS Agreement interpretation for database design
- 7. Two-layer data model
- 8. Clinical track data model
- 9. Cognito intake workflow and field mapping direction
- 10. Healthie post-enrollment workflow direction
- 11. Patient daily check-in and execution logging workflow
- 12. Data curation workflow and medical data assistants
- 13. Offshore clinical review boundary
- 14. Data source, normalization, provenance, and audit model
- 15. Suggested core schema modules
- 16. Security and compliance boundaries
- 17. Role and permission matrix
- 18. Public reporting and dashboard safeguards
- 19. GitHub repository setup and contractor operating rules
- 20. Phase 1 deliverables and Week 1 assignment
- 21. What not to do in Phase 1
- 22. Materials to share with Nathan
- 23. Additional content still needed
- 24. Acceptance criteria
- Appendix A. Proposed repository structure

1. Purpose and scope

This document defines the Version 2.0 data infrastructure direction for MetaFlow Health Inc. in connection with ACCESS-related operations. It is intended to guide database architecture, schema design, data provenance, reporting workflows, source-system boundaries, patient daily operations, data curation, offshore review boundaries, and future interoperability planning.

This is a Phase 1 technical onboarding brief and operating manual draft. It is not a final production SOP, legal compliance policy, clinical protocol, CMS API specification, or full EHR implementation plan.

- Primary audience: Nathan Ogden, technical contractor from Arc.dev.
- Secondary audience: MetaFlow internal stakeholders involved in technical, clinical governance, and data operations.
- Phase 1 goal: create a clear ACCESS-specific operational database foundation using synthetic data only.
- Phase 1 exclusion: do not build a full EHR, patient app, production CMS integration, or production PHI workflow in the first phase.

2. Executive summary

MetaFlow should not build this system as a simple patient intake database, CRM, or general research spreadsheet. The database should be designed as an ACCESS operational, outcomes, compliance, and evidence ledger.

The ACCESS-required layer should support beneficiary eligibility, alignment, unalignment, clinical track selection, care period management, OAP measure tracking, reporting windows, CMS submission status, provenance, and audit history. In parallel, MetaFlow will maintain an enhanced evidence layer for monthly labs, imaging reports, structural evidence, protocol versions, patient daily execution logs, physician review, adherence, safety signals, and future payer/investor evidence.

Cognito Forms and Healthie are important source/workflow systems, but neither should be treated as the final ACCESS data ledger. Cognito should be treated as the pre-enrollment intake and safety-screening source. Healthie should be treated as the post-enrollment patient/care platform. The ACCESS Core Database should normalize, validate, version, link, and audit data across systems.

Version 2.0 adds four major modules missing or underdeveloped in v0.1: patient daily check-in and execution logging, data curation workflow for future medical data assistants, offshore de-identified clinical review boundary, and a clearer source-of-truth / role-permission model.

3. Phase 1 development boundary

Phase 1 must remain synthetic-data-only and architecture-first.

- No real patient data, PHI, MBI, Medicare identifiers, CMS credentials, production credentials, Cognito admin credentials, Healthie admin credentials, secrets, or API keys should be used in GitHub or any local development environment.
- No real Cognito submissions, real Healthie patient records, real lab PDFs, real imaging PDFs, raw CMS files, or unredacted patient documents should be used in Phase 1.
- Development data should be synthetic. Any examples should use fake beneficiaries, fake labs, fake PROMs, fake imaging summaries, fake protocol versions, and fake CMS responses.
- Contractor access should be limited to repository materials, technical briefs, synthetic data, and test endpoints unless explicitly expanded later.
- A future production environment must be designed separately with formal security, compliance, access-control, backup, retention, and incident-response controls.

4. Design doctrine

- The core object is not simply a patient profile.
- The core object is a beneficiary-care-period-clinical-track-measure-submission-audit trail.
- Every reportable value must be traceable to source, collection date/time, method of collection, review status, reporting-window status, submission status, and audit history.
- Raw source data, normalized clinical observations, derived outcome calculations, and CMS submission payloads should be separated.
- ACCESS OAP reporting and MetaFlow enhanced evidence should be connected by beneficiary, care period, clinical track, and timeline, but should not be mixed into one undifferentiated table.
- Patient daily records are operational evidence and safety signals. They should not automatically become ACCESS OAP reportable data unless they satisfy the applicable ACCESS data collection, validity-window, provenance, and submission requirements.
- Medical data assistant output should enter the system as curated draft observations, not final clinical truth.
- Offshore clinical reviewers should receive de-identified or pseudonymized clinical packets only in Phase 1 and early operations.
- Phase 1 should produce an architecture memo, schema draft, data flow, synthetic workflow, and security boundary - not a full production system.

5. System map and source-of-truth model

The initial operating model should separate systems by function. This separation is important for security, auditability, future integrations, contractor access control, and patient-facing operations.

System / Layer	Role in Phase 1 architecture	Important boundary
Public website	Marketing and navigation layer. It may route prospective patients to intake.	Should not store clinical data or PHI.
Cognito Forms	Pre-enrollment intake source. Captures initial assessment, safety screening, uploaded files, participation readiness, and structured intake responses.	Use preview/test link and synthetic submissions for Nathan. No real submission data or admin access in Phase 1.
Healthie	Post-enrollment patient/care platform after physician approval and enrollment. Used for patient portal, secure communication, care plans, daily/ongoing workflow, and document upload as appropriate.	Healthie is not the ACCESS ledger. It should integrate with or feed the Core DB rather than replace it.
ACCESS Core Database	Operational source of truth for eligibility, alignment, care period, OAP measures, reporting windows, CMS submission status, provenance, normalized evidence, and audit trail.	This is the main architecture target for Nathan.
Encrypted document storage	Stores lab PDFs, imaging reports, clinical documents, PROM screenshots/version archives, and physician documents.	Relational DB should store metadata, hashes, version references, and links - not large files directly.
Lab / imaging sources	Quest/Labcorp/other labs, imaging centers, and report uploads provide evidence for observations.	Need source, collection date, report date, document link, normalization, and review status.
CMS ACCESS APIs	Future Eligibility, Alignment, Unalignment, and Data Reporting integration.	Schema should be API-ready even if production API integration is not built immediately.
Future HIE/FHIR/USCDI layer	Future interoperability path for structured health data exchange.	Plan now; implement later.

5.1 Proposed high-level flow

1. Website routes prospective patient to Cognito intake form.

2. Cognito captures intake, safety screening, and uploads. Phase 1 testing uses synthetic data only.
3. ACCESS Core DB normalizes intake data into identity, safety, clinical intake, document metadata, and workflow status tables.
4. MetaFlow internal reviewers perform initial review; Dr. Quinn or another authorized physician performs final clinical review where required.
5. Approved patients move into the enrolled-patient workflow, including Healthie setup as appropriate.
6. ACCESS eligibility/alignment workflow is tracked in the Core DB.
7. D0 baseline, monthly follow-ups, OAP measures, supplemental labs/imaging, protocol versions, daily execution logs, and safety signals are linked to the care period.
8. CMS reporting payloads and responses are stored with full provenance and audit history.
9. Daily patient records feed adherence, safety, and monthly review workflows, but do not automatically become OAP reportable observations.

5.2 Source-of-truth matrix

Data domain	System of capture	Operational source of truth	Notes
Pre-enrollment intake	Cognito Forms	ACCESS Core DB after normalization	Cognito stores source submission; Core DB stores normalized status, field mapping, and audit reference.
Patient identity / identifiers	Cognito or Healthie during workflow	ACCESS Core DB identity module with PHI controls	Identifiers should be separated from clinical observations and governed by least privilege.
Safety stop screening	Cognito Forms	ACCESS Core DB safety_screening module	Failed safety screening should route to not eligible / physician review status as appropriate.
Enrollment and care period	ACCESS Core DB	ACCESS Core DB	Care period is central to ACCESS reporting and evidence timeline.
Post-enrollment portal and messages	Healthie	Healthie for patient-facing workflow; Core DB for normalized status and references	Do not blindly duplicate all messages into the Core DB.
Daily execution and symptom records	Healthie or interim Cognito daily form	ACCESS Core DB daily operations module after ingestion	Used for adherence, safety review, and enhanced evidence.
OAP measure observations	Labs, PROM instruments, devices, Healthie/Cognito where applicable	ACCESS Core DB OAP ledger	Must include source, method, collection date/time, reporting-window status, review, and submission status.
Uploaded PDFs / imaging	Cognito, Healthie, direct upload, imaging centers	Encrypted object storage + Core DB metadata	Store file metadata and links in DB; not large files directly.
CMS API requests/responses	CMS APIs	ACCESS Core DB CMS integration module	No real CMS credentials or production responses in Phase 1.
Public dashboard aggregates	ACCESS Core DB analytics views	Approved reporting layer	Must respect de-identification, small-cell suppression, and CMS-derived distinctions.

6. ACCESS Agreement interpretation for database design

6.1 Beneficiary eligibility and alignment

The system should support a workflow where a potential beneficiary is screened, checked for ACCESS eligibility, and then categorized as ineligible, eligible for alignment, assigned to a control group, aligned to MetaFlow under a clinical track, or later unaligned if required.

- Track beneficiary identity and identifiers separately from clinical intake.
- Track requested clinical track and eligibility-check result.
- Store CMS response, control group status, alignment status, alignment date, unalignment status, unalignment reason, and audit history.
- Model eligibility/alignment separately from general intake data.

6.2 ACCESS APIs

The schema and backend should anticipate future connection to four ACCESS API categories: Eligibility, Alignment, Unalignment, and Data Reporting. Production API integration is not required in Week 1, but the database should be designed so these integrations do not require a rebuild.

6.3 Care period

Each aligned beneficiary should be linked to a care period, clinical track, baseline status, reporting schedule, current operational status, physician review status, and reconciliation-related status.

- Care period start date
- Care period end date
- Day 0 baseline status
- Active / paused / completed / unaligned status
- Clinical track
- Physician review status
- Reporting schedule
- Reconciliation-related status

6.4 OAP measure reporting

ACCESS is centered on Outcome-Aligned Payments, so the database must be built around OAP measure tracking. The system should not only store values. It should prove where each value came from, when it was collected, whether it was valid for the reporting window, and whether it was submitted correctly.

Required item for each OAP observation	Reason
Measure name and clinical track	Defines what is being reported and under which track.
Value and unit	Supports normalized reporting and analytics.
Baseline / quarterly / end-of-period / early success type	Controls reporting logic and expected timing.
Collection date/time and submission date/time	Needed for timing validation and audit.
Clinical validity-window status	Determines whether the observation is reportable.
Data source and method of collection	Supports provenance and auditability.
Source document, device, or system reference	Links the value to evidence.
Physician review status	Supports clinical governance.
CMS submission payload and response	Supports reporting and reconciliation.
Audit trail	Prevents silent overwrites and supports traceability.

6.5 Reporting windows

The database should support timing logic for baseline, quarterly, end-of-period, and early success reporting windows. It should calculate whether each value is reportable, late, missing, or out of window.

6.6 Audit and provenance

For ACCESS operations, provenance is not optional. Every reportable clinical value should have source, date, method, entry user, review user, modification history, submission status, and source evidence. Corrections should create new events rather than silently overwriting prior clinical values.

7. Two-layer data model

Layer	Purpose	Examples
ACCESS OAP Layer	CMS-required clinical track reporting, payment/reconciliation support, and compliance/audit readiness.	Eligibility, alignment, care period, OAP measures, clinical validity windows, CMS submissions, CMS responses, audit events.
MetaFlow Enhanced Evidence Layer	Internal clinical governance, patient management, outcome tracking, future investor/payer/research evidence.	Monthly lab panels, imaging reports, structural reversal evidence, protocol version history, physician notes, daily execution logs, symptom trajectory, adherence, red/yellow/green safety signals.

The two layers should be linked by beneficiary, care period, clinical track, and timeline, but should remain analytically and operationally distinct.

8. Clinical track data model

8.1 CKM / eCKM

For ACCESS reporting, the database should support structured tracking of measures such as blood pressure, LDL-C and other lipid values where applicable, weight, BMI, HbA1c, eGFR, and uACR where applicable.

For MetaFlow enhanced tracking, the database should also support creatinine, urine protein, urine albumin/creatinine ratio, urinalysis, inflammatory markers, liver function, renal function, lipid trajectory, blood glucose trajectory, and other clinically relevant lab values. Internal kidney-axis status should be tracked separately, such as normalization trajectory or reversal trajectory across kidney-related markers.

8.2 MSK

For ACCESS MSK reporting, the system should support standardized PROM-based measures. The database should store instrument name, instrument version, body site, raw item responses, calculated score, completion date/time, baseline/quarterly/end-of-period status, device/browser metadata where relevant, and final submitted score.

PROM wording, response options, scoring logic, and version should be preserved. MetaFlow enhanced tracking should include daily pain/function self-report, activity tolerance, sleep disruption, flare status, execution adherence, red/yellow/green symptom flags, and functional trajectory.

8.3 Structural evidence layer

MetaFlow also tracks structural evidence that may not be part of ACCESS OAP primary reporting but is important for internal evidence generation and future payer/investor reporting. This includes carotid ultrasound reports, plaque size/morphology, fatty liver ultrasound, kidney ultrasound, thyroid ultrasound, other imaging reports, before/after comparison, and physician interpretation.

9. Cognito intake workflow and field mapping direction

Nathan should review the Cognito form structure. Cognito should be described clearly as the pre-enrollment intake and safety-screening source system. He should receive a test/preview link and synthetic examples only, not real submissions or admin access in Phase 1.

Cognito content	Database destination / mapping direction
Identity and contact fields	beneficiaries / beneficiary_identifiers, with PHI controls and encryption strategy.
Safety stop screening	safety_screening, exclusion flags, review status, reason codes.
Participation readiness	participation_readiness or intake_assessments status fields.
Primary concerns and chronic disease history	intake_assessments and condition/history substructures.
Sex-specific branch fields	female_health_history / male_health_history or structured intake sub-tables.
Uploaded lab/imaging documents	clinical_documents and document_metadata; actual files stored in encrypted object storage.
Submission metadata	form_submission_events: submitted_at, form_version, source_system, raw payload reference.
Initial routing status	intake_review_status / physician_reviews / enrollment_decisions.

9.1 Rules for Nathan when reviewing Cognito

- Use test/preview link only in Phase 1.
- Do not enter real patient data into the test form.
- Do not request real submissions or PHI.
- Do not request owner/admin access unless an integration task is explicitly approved later.
- Map fields into normalized tables rather than treating Cognito as the final database.
- Capture form version and raw payload reference so future form changes remain auditable.

10. Healthie post-enrollment workflow direction

Cognito is the intake source and Healthie is the formal post-enrollment patient/care platform. Healthie should be introduced as the enrolled-patient workflow layer, not as the ACCESS Core Database.

Healthie role	Database implication
Patient portal after approval/enrollment	Healthie patient/account identifiers should link back to beneficiary and care_period records.
Secure messaging and operational care workflow	Messages/encounters may be referenced or summarized according to future integration and compliance rules.
Care plan / goals / metrics where used	Core DB should store normalized operational status and audit references, not blindly duplicate everything.
Possible webhook/API integration	Need future integration roadmap and clear source-of-truth rules.
Post-enrollment daily/ongoing workflow support	Daily logs may originate from Healthie, custom app, or another system; source system must be explicit.

Important source-of-truth rule: Healthie may be the formal patient-facing care platform, but the ACCESS Core Database should remain the normalized ledger for ACCESS reporting, clinical track/care period status, OAP observations, submission status, and audit history.

11. Patient daily check-in and execution logging workflow

After formal enrollment, patients should record daily execution and status updates through the post-enrollment patient/care platform. MetaFlow currently expects Healthie to serve as the primary post-enrollment patient

portal and care management workflow. If needed, a Cognito daily check-in form may be used as an interim structured source for daily records.

Daily patient records should not be treated as the ACCESS Core Database itself. They should enter the system as source data, then be normalized, reviewed, and linked to the beneficiary, care period, clinical track, and protocol version.

```
Patient enters daily record in Healthie / Cognito
-> raw ingestion
-> normalized daily logs
-> red/yellow/green safety rules
-> care team review queue
-> physician review if needed
-> monthly summary
-> MetaFlow Enhanced Evidence Layer
-> not automatically CMS OAP unless it meets ACCESS reporting rules
```

11.1 Patient-facing approach by phase

Phase	Patient-facing tool	Purpose
Phase 1	Healthie daily check-in / care task / form / journal. If insufficient, Cognito daily check-in form as interim structured source.	Avoid building a custom patient app before the data model, compliance workflow, and first operational evidence are stable.
Phase 2	Healthie/Cognito webhook, API, or export into ACCESS Core DB.	Build daily adherence dashboard, missed check-in alerts, red/yellow/green queue, monthly summaries, and protocol review packets.
Phase 3	Custom MetaFlow patient app if justified by funding, data, and operational SOP.	Daily protocol guidance, execution timer, symptom check-in, device integration, care team alerts, patient education.

11.2 Minimum daily check-in content

Category	Example fields	Database target
Execution log	Completed today's protocol; time completed; duration; body areas/steps completed; skipped steps; missed reason; unusual reaction.	patient_execution_logs, protocol_assignment_versions, daily_checkin_responses
Daily status and safety	New or worsening symptoms; pain change; fatigue; sleep; dizziness; shortness of breath; chest pain; fever; swelling; bleeding; skin reaction; GI reaction; urination change; other concern.	daily_status_logs, safety_signal_events, care_team_review_tasks
CKM/eCKM daily fields	Home BP if applicable; weight if applicable; blood sugar if applicable; urination changes; swelling; energy; sleep; medication changes.	track_specific_daily_logs, daily_status_logs. Manual BP is internal monitoring unless it meets ACCESS device/provenance rules.
MSK daily fields	Pain score 0-10; function today; walking/stairs/lifting/sitting tolerance; sleep disruption; flare yes/no; pain medication use.	track_specific_daily_logs, daily_status_logs. Formal ACCESS MSK reporting still requires standardized PROM instruments.
Medication / care-event change	Medication change; ER/urgent care/hospitalization; new diagnosis/procedure; doctor visit; adverse event.	clinical_event_logs, safety_signal_events, physician_review_tasks
Free-text notes	Patient comments or concerns.	daily_checkin_responses with review flags; not automatically reportable.

11.3 Patient Daily Operations Module

The database should support a Patient Daily Operations Module. The module should be designed for low friction because patients should be able to complete the daily check-in in a few minutes.

- daily_checkin_templates
- daily_checkin_items
- daily_checkin_submissions
- daily_checkin_responses
- raw_daily_checkin_submissions
- patient_execution_logs
- daily_status_logs
- track_specific_daily_logs
- safety_signal_events
- missed_checkin_events
- care_team_review_tasks
- patient_adherence_summaries
- monthly_patient_summaries
- protocol_assignment_versions
- patient_notifications

11.4 Reporting boundary for daily patient-entered data

Daily patient-entered data should primarily be used for operational monitoring, adherence tracking, safety surveillance, care team review, and MetaFlow enhanced evidence. It should not automatically become ACCESS OAP reportable data unless it meets the applicable ACCESS data collection, validity-window, provenance, and submission requirements.

Data type	Typical ACCESS OAP status	Primary use
Daily execution record	Usually not OAP reportable	Adherence and execution evidence.
Daily symptom change	Usually not OAP reportable	Safety monitoring and trend interpretation.
Daily pain/function	Internal daily trajectory; formal MSK reporting uses standardized PROMs	Early response signal and care team review.
Patient-entered manual BP	Internal monitoring unless device/provenance rules are satisfied	Safety monitoring and trend awareness.
Labcorp/Quest labs	Can become OAP/enhanced evidence when source and timing requirements are met	Structured outcome evidence.
Standardized PROMs	Can become MSK OAP when instrument/version/scoring/reporting rules are met	ACCESS reporting and MSK trajectory.
Imaging reports	Enhanced evidence rather than primary OAP in most cases	Structural reversal evidence and physician review.

12. Data curation workflow and medical data assistants

MetaFlow expects to use part-time medical data assistants to clean and normalize incoming clinical data. Data cleaning should not occur in spreadsheets, email attachments, messaging apps, or local files. All data curation work should occur inside a controlled workflow connected to the ACCESS Core Database.

```

raw source ingestion
-> cleaning task assignment
-> curated draft
-> quality control review
-> clinician or physician review where required
-> approved normalized observation
-> ACCESS OAP ledger or MetaFlow enhanced evidence layer
-> audit trail

```

Draft for architecture review and synthetic-data-only development. Do not upload PHI, MBI, credentials, or secrets.

Medical data assistants may help extract and normalize values from intake submissions, lab reports, imaging reports, PROMs, daily records, and uploaded documents. However, their cleaned data should initially be stored as draft observations, not final clinical truth. A separate review and approval process should determine when a value becomes part of the official operational record.

Stage	Example tables	Rule
Raw ingestion	raw_ingestion_records, raw_daily_checkin_submissions, source_document_records	Preserve source metadata and raw payload reference.
Cleaning task	data_curation_tasks	Assign work without exposing more data than necessary.
Curated draft	curated_observation_drafts, curated_document_summaries	Draft only; not final operational truth.
QC review	curation_review_events	Track reviewer, decision, comments, and corrections.
Clinical/physician review	clinical_review_tasks, physician_reviews	Required when clinical interpretation or reporting status is involved.
Approved observation	clinical_observations, oap_measure_observations, structural_evidence	Only approved records enter official ledger or evidence layer.
Audit	audit_events	Every view/edit/export/approval should be traceable.

12.1 Suggested curation statuses

- imported
- cleaning_pending
- curated_draft
- needs_clarification
- qc_passed
- clinician_review_required
- physician_review_required
- approved_normalized_observation
- locked_for_reporting
- submitted_to_CMS
- rejected_or_superseded

13. Offshore clinical review boundary

For Phase 1 and early operations, offshore clinical reviewers should not access real PHI, MBI, CMS-derived identifiable data, production credentials, or unredacted patient documents. Offshore clinical reviewers should receive only de-identified or pseudonymized clinical packets, using internal case IDs rather than direct patient identifiers.

The system should be designed so that Cognito and Healthie may contain patient-facing or operational PHI, while the ACCESS Core Database maintains the normalized reporting, evidence, and audit ledger. Any offshore clinical interpretation workflow should be separated from identity management, CMS data, raw document storage, and production reporting.

Offshore reviewers may see	Offshore reviewers should not see
Internal case ID / pseudonymous ID	Name, address, phone, email
Age band, sex, clinical track	Full DOB, Medicare ID / MBI, SSN
Structured lab values and trends	Raw CMS files or CMS-derived identifiable patient-level data
De-identified symptom summaries	Unredacted Cognito or Healthie profile
De-identified imaging summaries	Unredacted lab or imaging PDFs with identifiers
Execution adherence and safety flags	Production credentials, admin accounts, API keys
Protocol version reference and clinical interpretation packet	Direct patient messaging unless formally authorized later

The architecture should eventually support PHI redaction before offshore review, pseudonymous case IDs, field-level access controls, audit logs for every view/edit/export action, no local download by default, restricted document access, review status tracking, and approval gates before any curated value becomes reportable.

Guiding principle: minimum necessary access. Each user should see only the data required for the assigned role and no more.

14. Data source, normalization, provenance, and audit model

Data category	Raw source	Normalized record	Derived / operational use
Intake	Cognito raw submission	intake_assessments, safety_screening	Eligibility routing and physician review.
Labs	Lab PDFs, structured lab feeds, patient uploads	lab_results, lab_result_items, clinical_observations	OAP observations, enhanced evidence, trend analysis.
PROMs	PROM form responses	prom_responses, prom_scores	MSK OAP reporting and trajectory tracking.
Imaging	Uploaded reports / imaging center docs	imaging_reports, structural_evidence	Supplemental outcome evidence and physician review.
Daily execution	Healthie / Cognito daily check-in / future app	patient_execution_logs, daily_status_logs, track_specific_daily_logs	Adherence, safety flags, protocol response tracking.
Physician review	Healthie/notes/review workflow	physician_reviews, review_events	Approval, pause, escalation, enrollment decision.
Medical data curation	Assistant draft values and summaries	curated_observation_drafts, curation_review_events	Controlled cleaning before approval.
CMS data	CMS APIs / files	cms_api_requests, cms_api_responses, cms_data_submissions	Eligibility, alignment, reporting, reconciliation.

Each clinical observation should link to source metadata, source document/system reference, collection date/time, method of collection, entry/review users, version, validity-window status, and audit events.

15. Suggested core schema modules

The initial schema should be modular. Nathan should evaluate and improve this list rather than simply implementing it exactly as written.

Module	Example tables
Identity and access	beneficiaries, beneficiary_identifiers, users, roles, permissions, access_grants
Intake and screening	intake_assessments, safety_screening, participation_readiness, intake_review_status, form_submission_events
ACCESS workflow	eligibility_checks, alignments, unalignments, clinical_tracks, care_periods, enrollment_decisions
OAP reporting	oap_measure_definitions, oap_measure_observations, oap_reporting_windows, cms_data_submissions
CMS integration	cms_api_requests, cms_api_responses, cms_claims_files, cms_payment_files, recoupment_events
Clinical data	clinical_observations, lab_results, lab_result_items, prom_instruments, prom_responses, prom_scores, imaging_reports
MetaFlow evidence	structural_evidence, protocol_versions, protocol_assignment_versions, patient_execution_logs, daily_status_logs, track_specific_daily_logs
Patient daily operations	daily_checkin_templates, daily_checkin_items, daily_checkin_submissions, daily_checkin_responses, missed_checkin_events, patient_adherence_summaries,

	monthly_patient_summaries
Data curation	raw_ingestion_records, data_curation_tasks, curated_observation_drafts, curation_review_events, source_document_records
Documents	clinical_documents, document_metadata, document_versions, document_access_events
Governance	physician_reviews, clinical_review_tasks, red_yellow_green_events, safety_signal_events, care_team_review_tasks, audit_events
Dashboard / analytics	reporting_views, aggregate_outcomes, suppression_rules, public_report_exports

16. Security and compliance boundaries

- Synthetic/sample data only in Phase 1.
- No real patient data, PHI, MBI, CMS credentials, production credentials, or secrets in GitHub or local development environments.
- No real Cognito submissions, Healthie patient records, raw CMS files, or unredacted patient documents in Phase 1.
- Role-based access control and least-privilege permissions.
- Encrypted PHI fields and secure credential handling.
- Separate development, staging, and production environments.
- Encrypted object storage for clinical documents.
- Append-only or immutable audit-event design for clinical changes.
- Clear separation between PHI, normalized observations, raw documents, derived calculations, and public reporting data.
- Contractor access should be limited to repository, architecture documents, synthetic data, and test endpoints unless explicitly expanded later.
- No local download by default for future PHI workflows unless specifically approved and logged.
- All future production integrations must have explicit environment, secrets, backup, retention, and incident-response policy.

17. Role and permission matrix

The architecture should support role-based access from the beginning, even if Phase 1 only implements synthetic-data development. The following matrix is an initial planning model, not a final production policy.

Role	Primary purpose	Suggested Phase 1 access boundary
Ling / owner	Business, compliance, clinical-governance direction, final product decisions.	Repository owner/admin. No PHI in repository.
Nathan / contractor	Architecture, schema, synthetic workflow, repository setup, technical memo.	GitHub Write access; synthetic data and documentation only; no production credentials or PHI.
Initial technical tester / handoff support	Explain prior ACCESS technical testing and notes.	Read/Triage or temporary Write only for handoff; no secrets or PHI.
Future medical data assistant	Clean and normalize incoming clinical data.	Controlled data curation queue; draft observations only; access limited by task.
US clinical reviewer	Review cleaned observations, summaries, safety signals, and workflow status.	Access to operational clinical review views as authorized.
Dr. Quinn / physician reviewer	Final clinical review, eligibility/enrollment decision, pause/escalation where required.	Physician review dashboard, approved clinical documents, relevant patient records under production policy.
Offshore clinical interpretation team	De-identified TCM/system-pattern interpretation and protocol suggestion draft.	Pseudonymized clinical packets only; no PHI, MBI, raw CMS data, or unredacted documents.

Patient	Daily check-in, messages, uploads, care plan tasks.	Patient portal only; own records only.
Admin/finance	Billing, invoice, non-clinical operations.	No clinical PHI unless specifically required and authorized.

18. Public reporting and dashboard safeguards

MetaFlow expects to build an outcome dashboard later. The database should support aggregated reporting, but it must be designed to avoid exposing small-cell or identifiable patient information. Dashboard-oriented views should eventually support aggregation level, denominator, cell size, suppression-required flag, public-reportable flag, CMS-derived vs MetaFlow-derived distinction, and de-identification status.

- Do not publish patient-level data.
- Do not publish small cells or values that can reasonably identify a patient.
- Keep CMS-derived data clearly distinguished from MetaFlow-derived enhanced evidence.
- Public dashboards should use approved aggregate views, not raw operational tables.
- Early small-N evidence should be handled as de-identified case narratives or internal dashboards until public reporting thresholds and policies are clear.

19. GitHub repository setup and contractor operating rules

The private GitHub repository should function as a Phase 1 technical workspace. It should not become a data room for patient records, credentials, legal documents with sensitive information, or production operations.

Repository item	Purpose
/docs/architecture	Technical brief, architecture memos, diagrams.
/docs/regulatory	Public ACCESS Agreement link and technical references. Avoid sensitive CMS credentials or non-public data.
/docs/intake	Cognito test link, field mapping notes, synthetic examples.
/docs/healthie	Healthie workflow notes and future integration assumptions.
/docs/security	Role matrix, environment policy, secrets policy, access-control notes.
/docs/technical-handoff	Initial ACCESS testing notes and handoff materials without secrets.
/database/schema	Schema drafts, migration drafts, ERD outputs.
/sample-data	Synthetic intake, labs, PROMs, imaging summaries, daily logs, fake CMS responses.
/scripts	Development utilities using synthetic data only.

19.1 First GitHub issue

Suggested issue title: Phase 1: ACCESS Core Database Architecture Review and Technical Handoff.

The issue should cover review of the technical brief, current early-stage codebase/testing work, Cognito intake mapping, Healthie integration assumptions, ACCESS Core DB architecture, synthetic data workflow, data curation workflow, offshore/de-identified clinical review boundary, security/access-control plan, and Week 1-4 implementation roadmap.

20. Phase 1 deliverables and Week 1 assignment

20.1 First-month deliverables

1. Database Architecture Memo v1: recommended data model, database structure, data flows, source-of-truth rules, and security boundaries.

2. ACCESS Data Flow Diagram: intake -> safety screen -> eligibility check -> alignment -> care period -> baseline -> follow-up -> CMS reporting -> audit/reconciliation.
3. Initial PostgreSQL Schema Draft: core ACCESS tables and MetaFlow enhanced evidence tables.
4. Synthetic Data Workflow: fake beneficiaries, labs, PROMs, imaging, daily logs, protocol versions, and CMS responses.
5. OAP Measure Ledger Design: how reportable values are stored, validated, reviewed, and submitted.
6. Document Storage and Provenance Design: how lab PDFs, imaging files, clinical documents, PROM screenshots, and notes are stored and linked.
7. Patient Daily Operations Design: daily check-in templates, execution logs, safety flags, missed check-ins, care team queue, and monthly summaries.
8. Data Curation Workflow: assistant task queue, draft observations, QC, clinician/physician approval gates, and audit history.
9. Offshore De-identified Review Workflow: pseudonymous clinical packet design and access restrictions.
10. Security and Access-Control Plan: roles, permissions, credential handling, environment variables, audit logging, repository rules.
11. Future Integration Roadmap: CMS APIs, Cognito, Healthie, lab networks, FHIR/USCDI/HIE, and dashboards.

20.2 Week 1 assignment

1. Review this technical brief and the ACCESS Participation Agreement.
2. Review the Cognito test/preview form and field structure using synthetic data only.
3. Review notes from the initial ACCESS technical/database connection test.
4. Set up or improve the initial GitHub repository structure and Phase 1 issue if requested.
5. Identify major risks, missing pieces, and schema decisions.
6. Provide a written Week 1 technical assessment and proposed architecture direction.
7. Include Cognito intake mapping, Healthie assumptions, patient daily workflow, data curation workflow, offshore review boundary, and security/access-control recommendations.

21. What not to do in Phase 1

- Do not build this as a simple patient intake database.
- Do not build a full EHR in Phase 1.
- Do not build a custom patient app in Phase 1 unless separately approved.
- Do not use real patient data or PHI.
- Do not place credentials, secrets, API keys, or PHI in GitHub.
- Do not request real Cognito submissions or Healthie patient records.
- Do not treat labs, PROMs, documents, daily logs, and CMS submissions as the same type of data.
- Do not treat patient-entered daily records as ACCESS OAP truth by default.
- Do not allow data assistant cleaned values to become final truth without review/approval.
- Do not give offshore reviewers access to PHI, MBI, raw CMS data, or unredacted patient documents in Phase 1.
- Do not overwrite clinical values without audit history.
- Do not expose proprietary clinical decision logic or protocol selection rules in the technical repository.

22. Materials to share with Nathan

Material	How to share	Purpose
ACCESS Participation Agreement	Official public PDF link and/or repository copy under /docs/regulatory/.	Regulatory and reporting context.
ACCESS technical manual / API notes	Repository /docs/regulatory/ or /docs/api/.	API-ready schema planning.
This technical brief	PDF and editable DOCX in repository	Architecture direction and operating

Draft for architecture review and synthetic-data-only development. Do not upload PHI, MBI, credentials, or secrets.

	/docs/architecture/.	boundaries.
Cognito intake preview/test link	Markdown file with warning: synthetic data only.	Field mapping and intake workflow design.
Cognito field export or screenshots	Repository /docs/intake/.	Mapping form fields to tables.
Synthetic intake examples	Repository /sample-data/.	Testing without PHI.
Synthetic daily check-in examples	Repository /sample-data/.	Patient daily operations design.
Healthie workflow notes	Repository /docs/healthie/.	Post-enrollment platform integration planning.
Initial database/backend connection notes	Repository /docs/technical-handoff/.	Technical handoff from prior test work.
Role matrix and security notes	Repository /docs/security/.	Access control and workflow ownership.

23. Additional content still needed

1. Cognito form structure export: field names, field types, required/optional status, branch logic, upload fields, and form version.
2. Cognito test submission examples: at least one fake CKM/eCKM intake and one fake MSK intake.
3. Healthie workflow decision: exactly when a patient is created in Healthie, what data flows into Healthie, and what returns to the Core DB.
4. Daily check-in template: exact daily patient questions for CKM/eCKM and MSK, with red/yellow/green safety rules.
5. Source-of-truth matrix: confirm which system owns each data type in production.
6. Role and permission matrix: final version for Ling, Nathan, Dr. Quinn, data assistants, internal clinical reviewers, offshore reviewers, patients, future support staff, finance/admin users.
7. Data retention and deletion policy draft: especially for rejected intake records, withdrawn patients, test data, daily logs, and CMS-related records.
8. Environment policy: development, staging, production, secrets handling, migration policy, backup policy.
9. Synthetic data package: fake beneficiaries, fake labs, fake PROMs, fake imaging summaries, fake protocol versions, fake daily logs, fake curation tasks, and fake CMS responses.
10. Acceptance criteria: what Nathan must deliver by end of Week 1, Week 2, and end of Month 1.
11. Open compliance questions: items that need attorney/compliance review rather than contractor judgment.

24. Acceptance criteria

Phase 1 is successful if MetaFlow has a clear, secure, and implementation-ready architecture foundation without prematurely building a full EHR or exposing real patient data.

- Nathan has produced a Database Architecture Memo v1.
- The schema direction clearly separates ACCESS OAP reporting from MetaFlow enhanced evidence.
- Cognito, Healthie, ACCESS Core DB, object storage, labs/imaging, CMS APIs, and dashboard layers have clear source-of-truth boundaries.
- Patient daily check-in and execution logging are modeled as operational evidence, not automatic OAP truth.
- Medical data assistant cleaning is modeled as draft curation with QC and approval gates.
- Offshore clinical review is modeled with pseudonymized/de-identified packets only.
- Security, role-based access, audit, environment separation, and secrets handling are explicitly addressed.
- Synthetic data exists for intake, labs, PROMs, imaging, daily logs, curation tasks, protocol versions, and fake CMS responses.
- The repository is clean, private, documented, and free of PHI, credentials, secrets, and production data.
- The first version provides a credible path to future CMS API integration, Healthie/Cognito integration, lab data, FHIR/USCDI/HIE interoperability, and outcome dashboards.

Appendix A. Proposed repository structure

```
metaflow-access-data-infra/  
docs/  
  architecture/  
    MetaFlow_ACCESS_Data_Infrastructure_Brief_v2.0.pdf  
    MetaFlow_ACCESS_Data_Infrastructure_Brief_v2.0.docx  
  regulatory/  
    ACCESS_Participation_Agreement_Link.md  
    ACCESS_Technical_Manual_Notes.md  
  intake/  
    Cognito_Intake_Test_Link.md  
    Cognito_Field_Mapping_Notes_v0.1.md  
  healthie/  
    Healthie_Post_Enrollment_Workflow_Notes.md  
  security/  
    Role_Permission_Matrix_v0.1.md  
    Environment_and_Secrets_Policy_v0.1.md  
  technical-handoff/  
    Initial_ACCESS_Technical_Testing_Notes.md  
database/  
  schema/  
    schema_draft_v0.1.sql  
    erd_notes_v0.1.md  
sample-data/  
  synthetic_intake_ckm_example.json  
  synthetic_intake_msk_example.json  
  synthetic_lab_result_example.json  
  synthetic_msk_prom_example.json  
  synthetic_imaging_summary_example.json  
  synthetic_daily_checkin_example.json  
  synthetic_curation_task_example.json  
  synthetic_cms_response_example.json  
scripts/  
README.md  
.env.example  
.gitignore
```

Reference

Official ACCESS Participation Agreement PDF:

https://content.govdelivery.com/attachments/USCMS/2026/06/12/file_attachments/3682758/ACCESS%20Participation%20Agreement%20Final%2006112026_%20508.pdf