CDM and DRR questions

Below is a structured “shopping list” you can give your manager that captures every area where the ISDA team can accelerate or de-risk DRR delivery. Grouping is by lifecycle phase so you can pick and choose depending on budget and time.

1. Strategy & Scope

   • Confirm overall DRR vision vs. ISDA’s CDM roadmap and future regulatory plans

   • Prioritisation of jurisdictions and report types (e.g. EMIR-REFIT, CFTC Rewrite, JFSA, MAS2013, HKMA, ASIC)

   • Definition of MVP vs. long-term capabilities (trade, position, valuation, collateral, margin, post-trade events, etc.)

2. Regulatory Interpretation & Rulebooks

   • Authoritative interpretation notes for each rule set (reportability, fields, derivations, timelines)

   • Gap analysis between CDM data model and each regulator’s schema (ISO 20022, DTCC RDS, FPML, bespoke XML/CSV)

   • Maintenance process for “living” rulebooks—how changes flow from regulators into DRR code

3. CDM Data-Model Alignment

   • Mapping of regulator data items to CDM objects/attributes

   • Create/extend CDM primitives or enumerations where gaps exist (e.g. product templates, lifecycle events)

   • Validation rules in rosetta DSL or XText to guarantee model completeness before mapping stage

4. Technical Architecture & Tooling

   • Best-practice guidelines for using Rosetta DSL, code generators, and build plugins

   • Reference implementations for:

– Event-Scenario ingestion (FpML, CSV, Excel)

– Model enrichment pipelines (UPI, LEI, ISO10383 MIC, etc.)

– Mapping analytics & QA dashboards

• DevOps: recommended CI pipelines, versioning, release strategy for DRR artifacts

5. Mapping & Transformation Assets

   • Author or review mapping spreadsheets/XText specs for every jurisdiction+message pair

   • Provide sample Rosetta “projection” rules for ISO 20022 and DTCC RDS outputs

   • Reusable code for complex transformations (e.g. notional calculation, FX fixing, collateral aggregations)

6. Test-Case Library

   • Regulator-approved golden-source trade/valuation scenarios (simple, exotic, delegated, compressions)

   • Edge cases: multi-leg, mixed-asset baskets, novations, position transfers, stale data scenarios

   • JSON/XML expected-result files to enable regression packs and CI gates

7. Validation & Certification

   • Independent validation of rules vs. regulatory technical standards

   • Benchmarking of DRR output against existing vendor platforms or regulator test harnesses

   • Support for regulator “user-acceptance tests” (e.g. ESMA Conformance, CFTC validation tool)

8. Documentation & Training

   • Developer and business-analyst how-to guides, updated as model evolves

   • Workshops on Rosetta DSL, CDM semantics, and DRR governance processes

   • On-boarding materials for new banks or vendors joining the ecosystem

9. Governance & Change-Control

• Define steering-committee cadence, artefact versioning rules, and SLA for regulatory updates

• Contribution guidelines (GitHub flow, code reviews, semantic versioning)

• License/usage terms clarity for open-source vs. proprietary deliverables

10. Community & Industry Liaison

    • Engage other trade associations (AFME, FIA, ISLA) for cross-asset alignment

    • Coordinate feedback loops with TRs, SEFs, CCPs and vendors to minimise interpretation drift

    • Run periodic “code sprints” with industry participants to validate rules on real portfolios

11. Production Roll-out Support

    • Cut-over planning, dry-runs, and data-migration playbooks

    • Performance tuning and cloud cost optimisation of DRR pipelines

    • Post-go-live monitoring dashboards and support rota establishment

12. Future-Proofing & Innovation

    • Extension of DRR artefacts for new regimes (Basel CRR3, Sustainability, Digital Asset reporting)

    • Inter-operability with DLT or smart-contract platforms (e.g. synchronised reporting on-chain)

    • Analytics add-ons (data quality scoring, reconciliation, exception management)

How to use this list:
• Mark each bullet as “Need”, “Nice-to-have”, or “Internal”.

• Rank by regulatory deadlines and internal capacity.

• Draft a Statement of Work with ISDA team listing deliverables per quarter.

That should give your manager full visibility of every scenario where ISDA’s expertise could plug in.

Below is a much more granular “engagement plan” you can hand to the incoming ISDA CDM experts.

Think of it as a shopping list of concrete deliverables, questions, and pain-points you’ll want them to tackle with you. Adapt the parts that don’t apply to your scope.

A. PROJECT FOUNDATIONS

---

1. Use-Case Catalogue

   • Confirm the exact trade types, asset classes, and lifecycle events you intend to support in phase 1.

   • Deliverable ⇒ A signed-off table (trade type ✓, event ✓, target date).

2. Environment Set-up

   • Review our current repo (rosetta-source, examples, tests) and build chain (Maven, Rosetta-CLI, Python bindings).

   • Deliverable ⇒ Clean “reference build” script + container image that compiles CDM and runs unit tests on day 1.

3. Version & Upgrade Strategy

   • Which CDM tag (5.20, 6.0, nightly)? How will we absorb future releases?

   • Deliverable ⇒ Written policy + automated dependency bump pipeline.

B. DOMAIN & DATA MODELLING

---

4. Product Modelling Deep-Dive

• Walk through CDM representation of each product (e.g., Fixed/Floating IRS, Equity Swap, Repo).

• Map custom fields (if any) → CDM extensions or annotations.

• Deliverable ⇒ Product Modelling Specification per asset class, including any extension proposals.

1. Event & Workflow Semantics

   - Model each lifecycle step (Execute, Novate, Amend, Part-Transfer…) using CDM Event construct (Before/After states, lineage, primitives).
   - Validate against ISDA event taxonomy and sample files.
   - Deliverable ⇒ BPMN-style diagrams + concrete CDM JSON examples for every event type we need.

2. Static & Reference Data
   - Party, Account, Identifier, Asset static data objects; canonical code lists.
   - Strategy for golden-source sync vs. local cache.
   - Deliverable ⇒ Reference-data schemas + load scripts, approved by ops team.

C. DATA INGESTION & OUTBOUND MAPPING

---

7. Inbound Mapping (FpML, DTCC, CME, FIX, proprietary)

- Use Rosetta ingestion framework found in `rosetta-source/…/ingestions`.
- Write or review mapping transforms for each source schema → CDM.
- Edge-case handling (e.g., non-standard option payoffs).
- Deliverable ⇒ Mapping specification doc + transform code + unit tests with ISDA sample files.

1. Outbound / Interoperability
   - CDM → reg-report schema (EMIR REFIT, CFTC Rewrite) or settlement feeds.
   - JSON vs. Avro vs. Protobuf decision.
   - Deliverable ⇒ XSD/JSON-Schema definitions + sample payloads validated by regulator sandbox where possible.

D. CODE-GEN & FUNCTION LIBRARY

---

9. Rosetta DSL & Code Generation

- Extend or tweak functions in `rosetta-source/…/functions` (pricing, schedule generation, business-day adjustment).
- Set up multi-language build (Java, Scala, Python) with identical semantics.

• Deliverable ⇒ Up-to-date Rosetta source, passing regression tests, plus CI job that auto-generates language artifacts.

1. Performance & Memory Profiling
   • Process 100 k trade portfolio through "daily cycle" of events.
   • Tune JVM flags, data-structure reuse, or streaming APIs.
   • Deliverable ⇒ Benchmark report + performance budget and recommendations.

E. TESTING & CERTIFICATION

---

11. Conformance Test Packs

• Integrate ISDA test-pack XML/JSON files under `rosetta-source/.../test-pack`.

• Automate pass/fail in CI with coverage metrics.

• Deliverable ⇒ Green build plus dashboard showing % of official tests passed.

1. Golden-Copy Regression Suite
   • Create a "snapshotted" set of real trades/events; verify round-trip (Source → CDM → Source).
   • Deliverable ⇒ Regression repository + diff tool for future upgrades.

F. DEPLOYMENT & OPERATIONS

---

13. Service Layout

• Decide micro-service vs. monolith; API endpoints for ingest, query, event-submit.

• Deliverable ⇒ Architecture diagram + OpenAPI spec.

1. Observability & Audit Trail

   • Design immutable event log; integrate hash-chain / digital signature functions (`org.isda.cdm.hashing`).

   • Deliverable ⇒ Logging/Audit spec + working PoC.

2. Production Readiness

   • Disaster-recovery RTO/RPO, versioned schema migration strategy.

   • Deliverable ⇒ Operations runbook.

G. GOVERNANCE & COLLABORATION

16. Extension Contribution Process

- How we raise ISDA pull requests, sign CLA, align on naming.
- Deliverable ⇒ Internal checklist + calendar cadence with ISDA maintainers.

1. Training & Knowledge Transfer
   - Deep-dive workshops (Rosetta DSL, event modelling, test harness).
   - Pair-programming sessions on our repo’s tricky areas.
   - Deliverable ⇒ Recorded sessions + Q&A backlog in project wiki.

WHEN ENGAGING THE ISDA TEAM, PROVIDE:

- Current repo link + branch.
- List of *blocking* issues or “unknown-unknowns”.
- Target go-live date & intermediate milestones.
- Preferred communication channel (Slack, Teams, GitHub Issues).

EXAMPLE INITIAL QUESTION SET FOR ISDA SMEs

1. Does the current CDM fully cover equity total-return swaps with compounding fees? If not, where should we extend?
2. For partial novation of a cleared trade, which CDM Event field(s) hold the leftover quantity vs. transferred quantity?
3. What is the recommended hash algorithm and envelope for regulatory-grade audit trails?
4. Any known performance bottlenecks when generating payment schedules > 30 years?
5. How should we version control Rosetta DSL extensions to survive upstream CDM bumps?

Feel free to copy-paste this framework into a slide deck or Confluence page. Tailor rows that don’t apply, and add priority or owner columns to turn it into a living project plan.