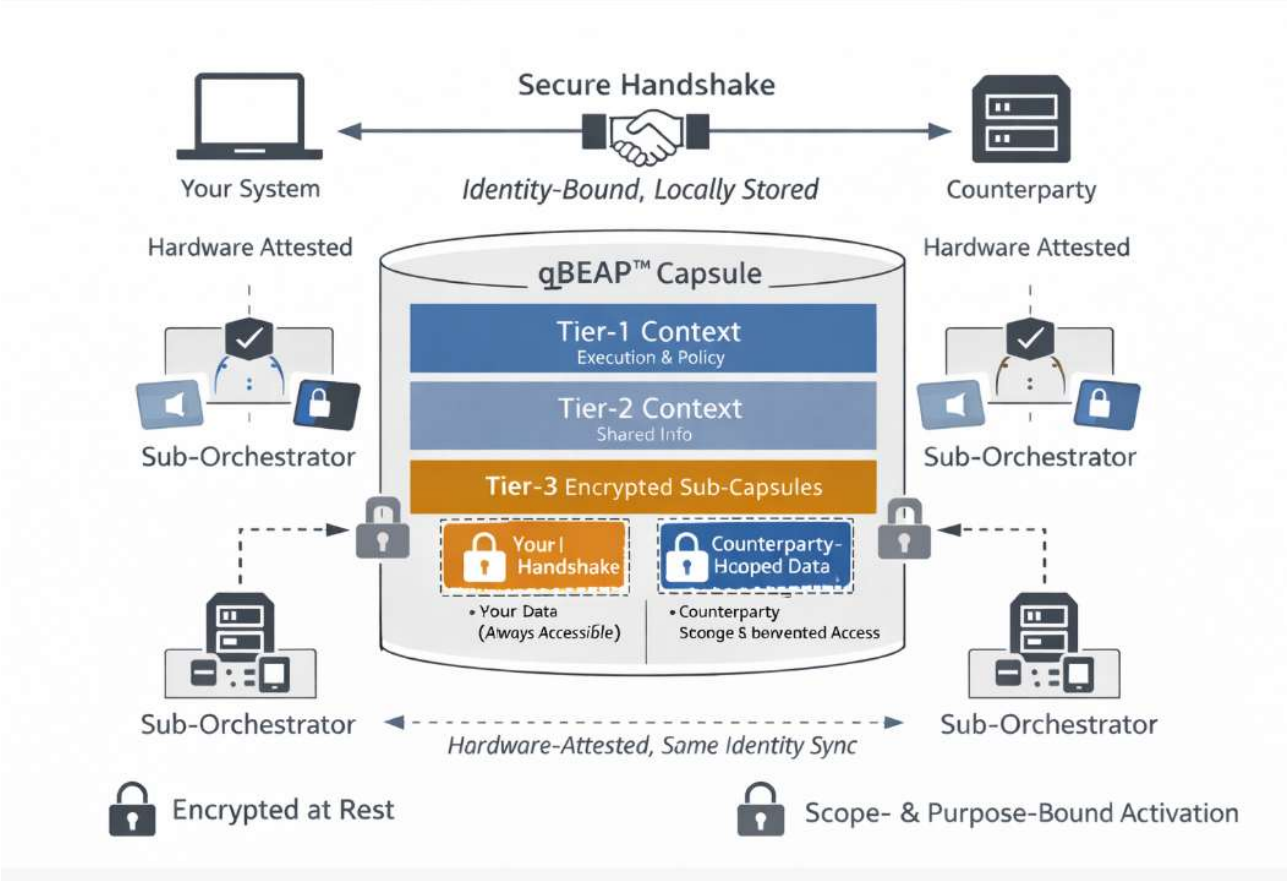

Annex A (Normative)

Expectation-Driven Handshake Synchronization, Attested Execution Environments, and Purpose-Bound Data Activation in BEAP™

A.1 Status and Scope

This annex is **normative**.

It defines mandatory **handshake semantics, storage rules, access boundaries, and synchronization constraints** for BEAP™-compliant implementations that support expectation-driven handshakes and handshake-scoped data.

This annex **does not modify or extend**:

- wire formats,
- capsule serialization formats,
- cryptographic primitives,
- transport mechanisms, or
- PoAE™ execution semantics

defined in the BEAP™, qBEAP™, and PoAE™ specifications.

This annex defines **behavioral, architectural, and security requirements** that MUST be honored by conforming implementations.

A.2 Applicability and Environment Constraints

The handshake model defined in this annex SHALL be supported **only** in **hardware-attested execution environments**.

An implementation MUST NOT enable handshake-scoped data storage, activation, or synchronization unless:

- the executing environment is hardware-attested, and
- the attestation is bound to a verifiable identity recognized by the BEAP™ trust model.

Non-attested environments MAY participate in BEAP™ interactions but SHALL NOT use the handshake-scoped mechanisms defined in this annex.

A.3 Handshake as an Operational Boundary

In BEAP™, a handshake SHALL represent a **bounded, identity-anchored cooperation state** shared between two parties.

A handshake state SHALL be:

- identity-bound,
- explicitly consented,
- cryptographically verifiable,
- revocable,
- and confined to attested execution environments.

The handshake SHALL function as a temporary operational contract governing:

- which data classes MAY exist within exchanged capsules,
- which expectations apply,
- which encrypted data regions MAY be accessed,
- and under which declared purposes activation MAY occur.

All shared context SHALL be strictly scoped to the lifetime of the handshake.

A.4 Local Handshake Persistence

Each participating party SHALL store the handshake state **locally**.

Local handshake persistence:

- SHALL be encrypted at rest,

- SHALL be cryptographically bound to the handshake identifier,
- SHALL be bound to the attested device or environment,
- SHALL NOT constitute a global session state,
- SHALL NOT require continuous online availability of the counterparty.

Local persistence SHALL NOT relax capsule-local determinism or PoAE™ policy enforcement.

A.5 Expectation-Driven Synchronization

A.5.1 Expectations as Declarative Signals

An **expectation** SHALL be a declarative statement describing information that MAY be required for a handshake to transition into an active operational state.

An expectation:

- SHALL NOT mandate immediate disclosure,
- SHALL NOT imply processing,
- SHALL define requirements, not actions.

An expectation MAY specify:

- attribute identifiers,
- data class (e.g. PII, business-critical, informational),
- purpose identifiers,
- requirement level,
- additional constraints (e.g. jurisdiction, format, validation state).

A.5.2 Iterative Capsule Synchronization

Handshake completion SHALL occur through **iterative capsule exchange**.

Capsules exchanged during synchronization MAY contain:

- capsule-local deterministic execution context,
- shared informational references,
- encrypted handshake-scoped data regions.

Sensitive data SHALL NOT be required to be processed solely to satisfy expectation negotiation.

A.6 qBEAP™ Capsule Structure and Encrypted Regions

A qBEAP™ capsule used under an active handshake MAY contain **multiple encrypted regions**, each with **distinct access semantics**.

A.6.1 Capsule Regions

A capsule MAY conceptually contain:

Region 1 — Tier-1 Context
Capsule-local deterministic execution context and policy references.
This region SHALL be the sole authority for execution and automation decisions.

Region 2 — Tier-2 Context
Shared informational and non-sensitive material. Augmented Overlay Annotations and support context eg Knowledgebase

Region 3 — Handshake-Scoped Encrypted Sub-Capsules
One or more encrypted sub-capsules containing handshake-scoped data such as PII, sensitive data, Augmented Overlay support for sensitive operations

---

A.6.2 Handshake-Scoped Encrypted Sub-Capsules

Handshake-scoped data SHALL be stored exclusively within **encrypted qBEAP™ sub-capsules**.

Each such sub-capsule:

- SHALL be encrypted at rest and in transit,
- SHALL be cryptographically bound to the handshake identifier,
- SHALL be isolated from execution context,
- SHALL NOT influence execution logic,
- SHALL be inaccessible without explicit activation.

Multiple encrypted sub-capsules MAY coexist within a single qBEAP™ capsule.

A.7 Access Semantics and Purpose-Bound Activation

Processing or access to handshake-scoped data SHALL be permitted only when **all** of the following conditions are met:

- a matching expectation exists,
- a declared purpose identifier applies,
- policy conditions are satisfied,
- valid consent is present,
- the requesting environment is hardware-attested.

Each party:

- SHALL be able to decrypt and access **its own contributed handshake-scoped data** at any time.
- SHALL NOT gain unrestricted access to counterparty-contributed data.

Access to counterparty-provided handshake data:

- SHALL be scope-bound,
- SHALL be purpose-bound,
- SHALL be denied by default.

Transport or storage of encrypted handshake-scoped data SHALL NOT constitute processing.

A.8 Sub-Orchestrator Synchronization

Handshake-scoped data MAY be synchronized between **sub-orchestrators** operating under the **same identity**, provided that:

- each sub-orchestrator executes in a hardware-attested environment,
- attestation confirms equivalence of trust level,
- the identity binding is verifiable and unchanged.

Synchronization MAY include:

- an empty qBEAP™ capsule,
- the handshake identifier,
- all encrypted handshake-scoped sub-capsules.

Synchronization SHALL NOT relax access controls or activation rules defined in this annex.

Sub-orchestrators operating under different identities or without hardware attestation SHALL NOT receive handshake-scoped data.

A.9 Permitted Data Introduction Paths

BEAP™ SHALL permit two normative mechanisms for introducing sensitive data into a handshake-scoped cooperation state:

1. **Expectation fulfillment via explicit data request**
2. **Handshake-anchored encrypted data presence**

In both cases, data SHALL remain inactive until purpose-bound activation occurs.

A.10 Security, Privacy, and Determinism Properties

The model defined in this annex:

- restricts processing to explicitly declared purposes,
- prevents implicit or ambient data access,
- preserves auditability and traceability,
- enforces attestation-bound trust boundaries,

- maintains strict determinism.

Handshake revocation SHALL immediately affect future access and synchronization without altering historical integrity.

A.11 Conclusion

This annex normatively defines a **hardware-attested, identity-bound handshake model** in which:

- handshake state is stored locally and encrypted at rest,
- multiple encrypted data regions coexist within qBEAP™ capsules as part of a handshake,
- access is strictly scope- and purpose-bound,
- synchronization is limited to attested environments under the same identity.

This model enables secure, low-friction cooperation without introducing hidden state, implicit trust, or non-deterministic behavior.

License Notice

This annex (**Annex A**) is an integral part of the **WRDesk™ specification**.

It is licensed under the **same license terms** as **WRDesk™**, **BEAP™**, and **PoAE™**,
as defined in the main repository **README** and accompanying license files.