FamilyOS Technical Privacy Appendix

Purpose

This appendix explains why FamilyOS can be more defensible than a generic AI wrapper: the product is built around governed family memory, consent, provenance, and safe continuity boundaries.

Core architecture

VISUAL FLOW / DIAGRAM BLUEPRINT

```
flowchart TD
    A[Inputs: voice, text, photos, docs, interviews] --> B[Ingestion + classification]
    B --> C[Memory object store]
    C --> D[Consent + policy engine]
    C --> E[Provenance/source trail]
    C --> F[Family graph]
    D --> G[Retrieval authorization]
    E --> G
    F --> G
    G --> H[Care Continuity outputs]
    G --> I[Family assistant answers]
    G --> J[Persona continuity outputs]
    G --> K[Device/robot context API]
```

Memory object model

Each memory should be stored as a governed object, not an unbounded chat transcript.

Field	Example
memory_id	`mem_2026_00123`
person	Courtney, Robert, family member, caregiver
subject	food preference, story, routine, value, calming cue
source	voice note, text, interview, document
confidence	high / medium / low
privacy	private / family / care circle / partner-approved
consent owner	person who gave permission
allowed uses	care, family answer, story, persona, device context
expiration/review	date or event-based review
delete/edit path	clear revocation route

Consent flow

VISUAL FLOW / DIAGRAM BLUEPRINT

```
sequenceDiagram
    participant U as Family/user
    participant OS as FamilyOS
    participant P as Policy engine
    participant R as Retriever
    U->>OS: Add memory
    OS->>U: Ask who can use it and for what
    U->>P: Set consent/access rules
    P->>OS: Attach policy to memory object
    R->>P: Request memory for answer/output
    P->>R: Approve/deny/filter
    R->>U: Return allowed answer with source boundary
```

Persona continuity safety

FamilyOS should use grief-safe language and avoid resurrection claims.

Allowed framing:

- permissioned traces
- voice, values, stories, preferences, humor, likely responses
- clearly labeled simulated continuity
- family-governed access
- edit/delete/revoke rights

Avoid:

- “bring them back”
- “live forever” as literal claim
- legal/medical/financial authority
- pretending the simulation is conscious
- changing a deceased person’s values without governance

Care Continuity output classes

Output	Audience	Safety boundary
Care handoff summary	family/caregiver	not medical advice
Preference card	aides/care team	source-labeled, editable
Story prompt	family	consented sharing only
Calming cue list	family/caregiver	non-clinical supportive cues
Family answer	approved family member	source + confidence where needed
Persona trace	approved family	explicitly labeled simulation

Privacy/security posture

Minimum investor-grade expectations:

Layer	Requirement
Data ownership	family-owned/exportable data
Access control	person, role, and use-specific permissions
Audit trail	who accessed what and why
Encryption	at rest and in transit
Deletion	clear delete and revocation path
Data minimization	collect only needed context
Training boundary	no family data used for general model training without explicit opt-in
Sensitive domains	medical/legal/financial claims blocked or routed to professionals

Device/robot portability concept

The long-term platform value is that FamilyOS becomes a governed context layer for future AI endpoints.

VISUAL FLOW / DIAGRAM BLUEPRINT

```
flowchart LR
    A[FamilyOS policy + memory layer] --> B[Phone assistant]
    A --> C[Home robot]
    A --> D[Smart speaker]
    A --> E[Vehicle assistant]
    A --> F[Appliance / smart home]
    A --> G[Care provider dashboard]
```

The device asks for context; FamilyOS decides what can be shared.

Example: robot context request

VISUAL FLOW / DIAGRAM BLUEPRINT

```
sequenceDiagram
    participant Robot as Home robot
    participant API as FamilyOS API
    participant Policy as Consent engine
    participant Mem as Memory graph
    Robot->>API: Request context: "How should I greet Grandpa?"
    API->>Policy: Is robot allowed this memory class?
    Policy->>Mem: Retrieve approved greeting preferences
    Mem->>API: Returns source-labeled memory objects
    API->>Robot: Approved greeting + restrictions
```

Safety gates before launch

- [ ] human-readable privacy policy
- [ ] consent and revocation UX
- [ ] export/delete function
- [ ] role-based access controls
- [ ] audit logs for sensitive retrieval
- [ ] clear non-medical/non-legal language
- [ ] persona labeling and grief-safe copy review
- [ ] security review before external pilots
- [ ] pilot participant informed-consent packet
- [ ] incident response plan

Technical moat summary

FamilyOS becomes defensible if it owns:

1. family memory object grammar
2. consent/provenance graph
3. care-continuity workflows
4. family governance UX
5. cross-device context permissions
6. grief-safe persona continuity patterns

The product should not compete only on model quality. It should compete on **trust architecture**.

Investor-draft strategic material. Financials are planning estimates pending pilot validation, legal/privacy review, and accountant review.