

Blame & Claim Flow Architecture

All active flows — V1 (legacy) · V2 · V3 · V4 · V5 — with roles, route prefixes, and step sequences.

Legend & Roles

Roles by Flow		
Flow	Blame actor	Claim/Review actor
V1 / V2 (user)	USER	USER + EXPERT
Expert panel	EXPERT FIELD_EXPERT	DAMAGE_EXPERT
V2 expert-init	FIELD_EXPERT	FIELD_EXPERT
V3	FIELD_EXPERT	FIELD_EXPERT
V4	FILE_MAKER	FILE_REVIEWER
V5	FILE_MAKER	FILE_REVIEWER + FINANCIAL_EXPERT

Route Prefixes		
Flow	Blame prefix	Claim / Review prefix
V1 (legacy)	blame-request-management/	claim-request-management/
V2 user	v2/blame-request-management/	v2/claim-request-management/
V2 expert-init	v2/expert-initiated/blame-request-management/	v2/expert-initiated/claim-request-management/
V2 expert panel	v2/expert-blame/	v2/expert-claim/
V3	v3/expert-initiated/blame-request-management/	same controller
V4	v4/file-maker/blame-request-management/	v4/file-reviewer/blame-request-management/
V5	v5/file-maker/blame-request-management/	v5/file-reviewer/blame-request-management/

Flow V1 / V2 — User-Initiated Blame & Claim

The canonical user flow. First party fills blame info and invites second party (third-party insurer). A blame expert optionally reviews. After blame is complete, the damaged party opens a claim which a damage expert prices. V1 = legacy routes · V2 = refactored same logic, new route prefix.

Blame Phase	
Actor:	USER Guard: GlobalGuard
1	Create request POST /
↓	
2	Blame confession POST /blame-confession/:id
↓	
3	Initial form (plate / VIN) POST /initial-form/:id or /initial-form-vin/:id
↓	

4 Details: location · voice · description

POST /add-detail-location, /upload-voice, /add-detail-description

↓

5 Add second party (sends SMS invite)

POST /add-second-party/:phone/:id/:route

↓

6 Second party fills same steps 2–4*Second party logs in on their own device*

↓

7 Sign (both parties)

PUT /sign/:id

↓

8 [Optional] Expert blame review

via v2/expert-blame – assign → reply/resend

Expert assigns case, may request resend, submits verdict

↓

9 Blame COMPLETED*Guilty party determined***Claim Phase**Actor: **USER** (damaged party) + **DAMAGE_EXPERT****1 Create claim from blame**

POST v2/claim-request-management/create-from-blame/:blameId

↓

2 Select outer parts

PATCH /select-outer-parts/:claimId

↓

3 Select other parts + bank info

PATCH /select-other-parts/:claimId

↓

4 Upload documents

POST /upload-document/:claimId

↓

5 Capture parts (photos + angles)

POST /capture-part/:claimId

↓

6 Car capture (walk-around video)

PATCH /car-capture/:claimId

→ WAITING_FOR_DAMAGE_EXPERT

↓

7 Damage expert reviews & prices

via v2/expert-claim – assign → reply / price-drop

↓

8 User signs pricing / objects / rates

PUT /owner-insurer-approval/sign or /objection or /user-rating

Flow V2 — Expert-Initiated (In-Person)

A **FIELD_EXPERT** at the accident scene fills every step on behalf of both parties using the same endpoint bodies as V2 user flow, just under the **v2/expert-initiated/** prefix. First registered party = guilty. No waiting-for-expert phase; blame is finalized with guilt decided immediately after signatures.

Blame Phase

Actor: **FIELD_EXPERT** Guard: LocalActorAuthGuard

- 1 **Create (creationMethod=IN_PERSON)**
POST v2/expert-initiated/blame-request-management/
↓
- 2 **Send & verify party OTP (guilty)**
POST /send-party-otp/:id → /verify-party-otp/:id
↓
- 3 **[if CAR_BODY] Car body form**
POST /car-body-form/:id
↓
- 4 **Run inquiries (initial form)**
POST /run-inquiries/:id
↓
- 5 **Details: location · voice · description (guilty)**
POST /add-detail-location, /upload-voice, /add-detail-description
↓
- 6 **Add second party (no SMS)**
POST /add-second-party/:phone/:id/
↓
- 7 **OTP + inquiries + details (damaged)**
↓
- 8 **Sign FIRST, then Sign SECOND**
PUT /sign/:id (called twice)
↓
- 9 **Accident fields → Blame COMPLETED**
POST /accident-fields/:id
No expert review queue — blame closes immediately

Claim Phase (same expert)

Actor: **FIELD_EXPERT**

- 1 **Create claim from blame**
POST v2/expert-initiated/claim-request-management/create-from-blame/:blameId
↓
- 2 **Select outer + other parts**
PATCH /select-outer-parts, /select-other-parts
↓
- 3 **Upload documents**
POST /upload-document/:claimId
↓
- 4 **Capture parts**
POST /capture-part/:claimId
↓

5 Car capture (walk-around video)

PATCH /car-capture/:claimId
→ WAITING_FOR_DAMAGE_EXPERT

↓

6 Damage expert reviews (normal flow)

via v2/expert-claim

Flow V3 — Expert-Initiated (Reorganised Steps)

Same actor as V2 (FIELD_EXPERT), same end result, but the company reorganised the step order: all party narrative steps (confession, inquiries, voice, location, description, sign) come first, then the damage-assessment steps (accident fields, documents, part selection, capture, video) follow in a single dedicated pass. Blame + Claim work in a single controller: [v3/expert-initiated/blame-request-management/](#).

Actor: **FIELD_EXPERT** Guard: LocalActorAuthGuard

PARTY STEPS (BLAME)

1 Create (IN_PERSON)

POST /

↓

2 OTP send + verify (guilty)

POST /send-party-otp → /verify-party-otp

↓

3 [CAR_BODY] Car body form

POST /car-body-form/:id

↓

4 Run inquiries (guilty + auto-claim)

POST /run-inquiries/:id

↓

5 Location · description · voice (guilty)

↓

6 Sign (guilty)

PUT /sign/:id

↓

7 OTP + inquiries + details (damaged)

THIRD_PARTY only; CAR_BODY skips 7–8

↓

8 Sign (damaged)

PUT /sign/:id

DAMAGE ASSESSMENT STEPS

9 Accident fields

POST /accident-fields/:id

↓

10 Get linked claim ID

GET /claim-id/:requestId

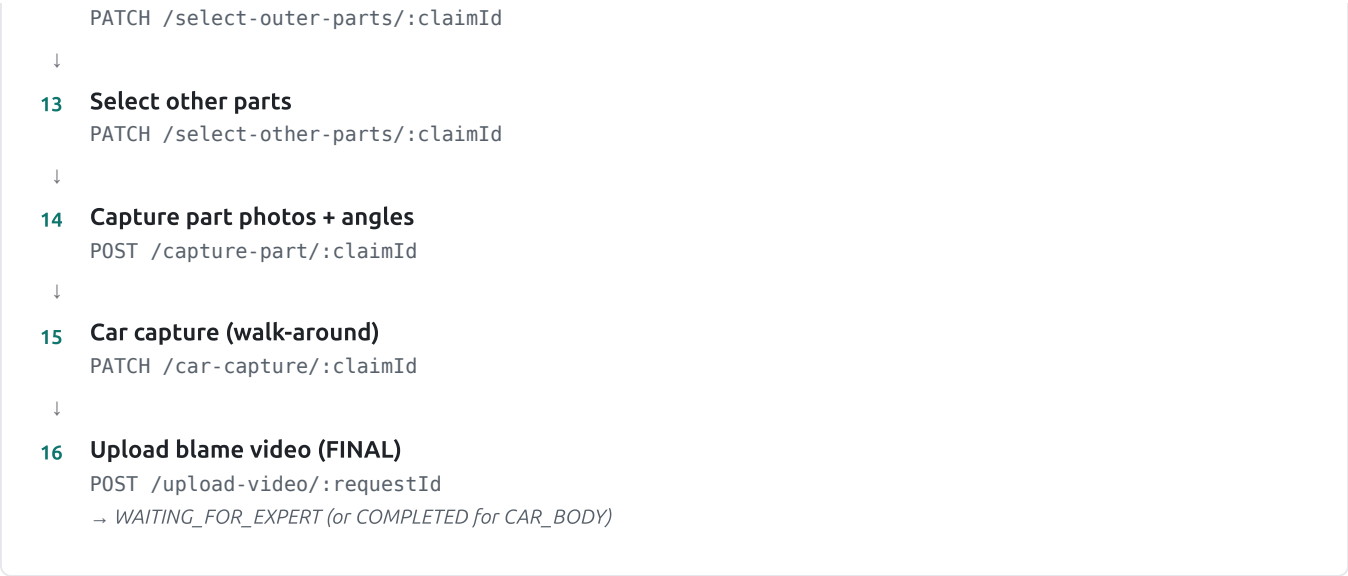
↓

11 Upload documents (licences, car cards)

POST /upload-document/:claimId

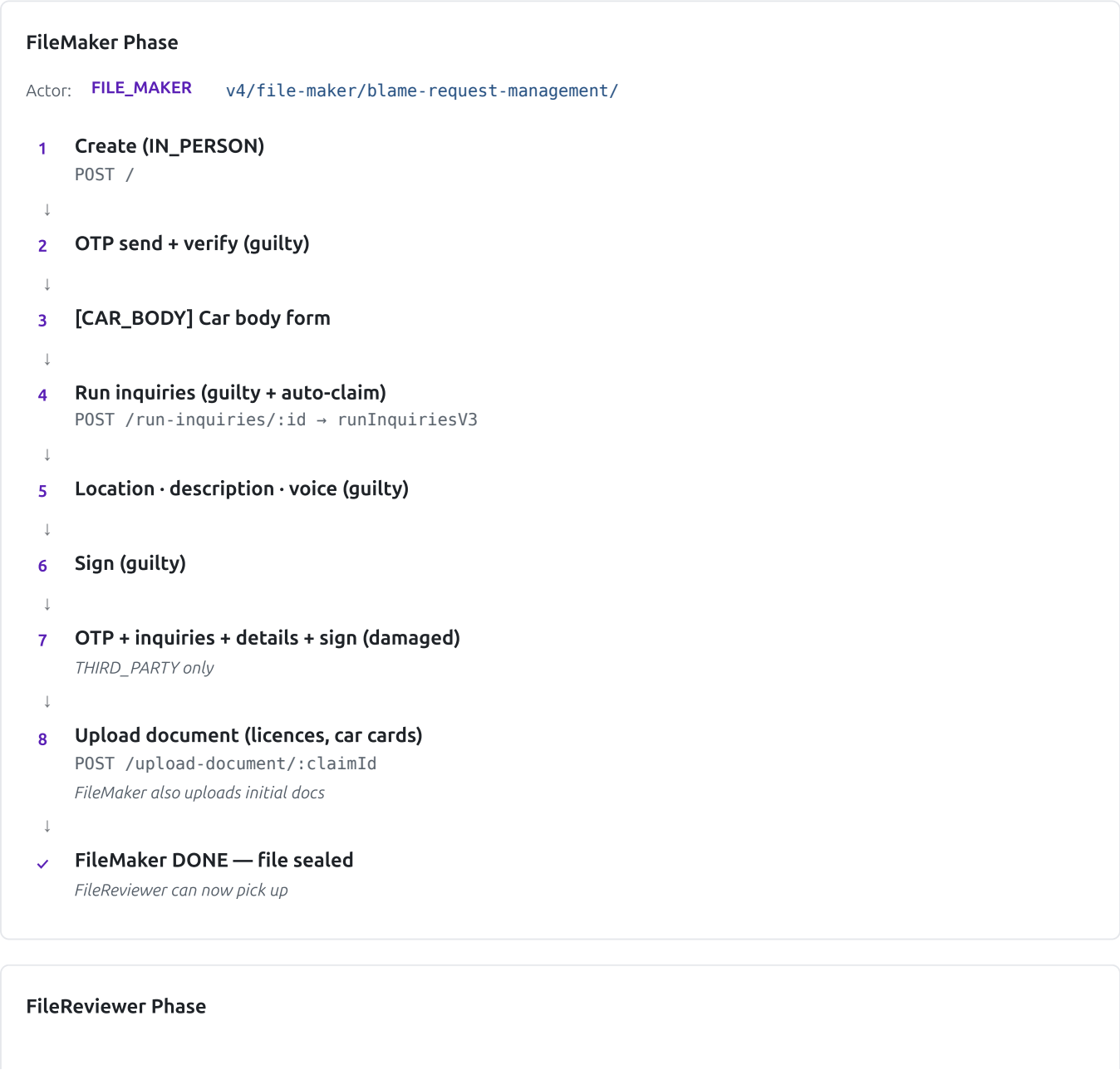
↓

12 Select outer parts



Flow V4 — Split Roles: FileMaker + FileReviewer

Two people handle the same V3 work. **FileMaker** handles party narrative (OTPs, inquiries, details, signatures). **FileReviewer** handles the damage assessment pass. Upload-video at the end is a no-op (blame already COMPLETED by car-capture).



Actor: **FILE_REVIEWER** [v4/file-reviewer/blame-request-management/](#)

- 1 Get linked claim ID**
GET /claim-id/:requestId
↓
- 2 Accident fields**
POST /accident-fields/:requestId
↓
- 3 Capture requirements lookup**
GET /capture-requirements/:claimId
↓
- 4 Upload document (chassis / engine)**
POST /upload-document/:claimId
↓
- 5 Select outer parts**
PATCH /select-outer-parts/:claimId
↓
- 6 Select other parts**
PATCH /select-other-parts/:claimId
↓
- 7 Capture part photos + angles**
POST /capture-part/:claimId
↓
- 8 Car capture — FINAL FileReviewer step**
PATCH /car-capture/:claimId
claim → WAITING_FOR_DAMAGE_EXPERT · blame → COMPLETED
↓
- upload-video — no-op**
POST /upload-video/:requestId (blame already COMPLETED)

Flow V5 — Same as V4 + Financial Expert Approval Gate

FileMaker steps are identical to V4. The only difference is the tail of the flow: after the damage expert completes their review, the claim moves to **WAITING_FOR_FILE_MAKER_APPROVAL** (via a **FINANCIAL_EXPERT** step) before fanavaran submission.

Actors: **FILE_MAKER** **FILE_REVIEWER** **FINANCIAL_EXPERT (approval)**

FILEMAKER STEPS (IDENTICAL TO V4)

Same sequence under [v5/file-maker/blame-request-management/](#) — no changes.

1–8 Identical to V4 FileMaker

See V4 left column above

↓

✓ FileMaker DONE — file sealed

FILEREVIEWER + APPROVAL TAIL

1–8 Identical to V4 FileReviewer (steps 1–8)

[v5/file-reviewer/blame-request-management/](#)

↓

9 Damage expert reviews claim

via [v2/expert-claim](#)

↓

10

Claim → WAITING_FOR_FILE_MAKER_APPROVAL

(instead of direct fanavaran submit)

↓

11

FileMaker approves or rejects

POST v5/file-maker/claim-approval/approve/:claimId

POST v5/file-maker/claim-approval/reject/:claimId

↓

✓

Fanavaran submission

Expert Review Panels (shared across V1–V5)

Expert Blame Panel — v2/expert-blame/

Actor:

EXPERT

FIELD_EXPERT

Method	Route	Purpose
GET	/	List blame cases for review
GET	/:id	Case details
POST	/:id/assign	Assign & lock case
PUT	/reply/submit/:id	Submit verdict
PUT	/reply/resend/:id	Request document resend
PUT	/reply/inPerson/:id	Submit in-person visit verdict
GET	/report/unified-file-statuses	Status catalog & counts

Expert Claim Panel — v2/expert-claim/

Actor:

DAMAGE_EXPERT

FIELD_EXPERT

FILE_REVIEWER

FILE_MAKER

Method	Route	Purpose
GET	/requests	List claim queue
GET	/request/:id	Claim details
POST	/assign/:id	Assign & lock
PUT	/reply/submit/:id	Submit damage assessment
PUT	/reply/resend/:id	Request resend
GET/PUT	/request/:id/price-drop	Price drop calculation
PATCH	/validate-factors/:id	Validate repair factors
PATCH	/:id/visit	Request in-person visit

Flow Comparison at a Glance

Dimension	V1 / V2 User	V2 Expert-Init	V3	V4	V5
Who fills blame?	Each party on own device	FIELD_EXPERT for both	FIELD_EXPERT for both	FILE_MAKER	FILE_MAKER
Who does assessment?	User, then expert reviews	FIELD_EXPERT	FIELD_EXPERT (same session)	FILE_REVIEWER	FILE_REVIEWER
Expert blame review?	Optional queue	No — immediate COMPLETED	WAITING_FOR_EXPERT after video	COMPLETED after car-capture	COMPLETED after car-capture
Claim review	DAMAGE_EXPERT	DAMAGE_EXPERT	DAMAGE_EXPERT	DAMAGE_EXPERT	DAMAGE_EXPERT → FileMaker approval

Upload-video step	Yes (first party)	Optional (not in V2 mirror seq.)	Yes — last step → WAITING_FOR_EXPERT	No-op (blame done by car-capture)	No-op (blame done by car-capture)
Blame route prefix	v2/blame-request-management/	v2/expert-initiated/blame.../	v3/expert-initiated/blame.../	v4/file-maker/blame.../	v5/file-maker/blame.../
Assessment route prefix	v2/claim-request-management/	v2/expert-initiated/claim.../	same v3/.../blame.../ controller	v4/file-reviewer/blame.../	v5/file-reviewer/blame.../

Made by Sepehr (Backend's Lord)