AI Personal CFO
v3 — the forward ledger

A second design with a different premise: the app is not a set of dashboards you read, it is one queue of decisions you work through. Same colours, down to the hex.

Against: `docs/design/design-language-v2.html` and `docs/design/AI_Personal_CFO_redesign.pdf` — v2, which answers *how am I doing?*
Mockups: `docs/design/redesign-v3/mockups/` · rendered to `shots/`.
Palette unchanged. `v3-screens.css` copies v2's token block byte for byte, and no rule below it contains a literal colour — only `var(--…)`. What differs is structure, density, and what the app asks of you.
No application code was changed to produce this.

The premise

v2 says: every screen should answer one question, and the app is organised by subject — Today, Money, Plan, Ask, You. You go to the screen that holds your question, and it leads with the figure that answers it. It is calm, it is legible, and the user does the choosing.

v3 says: a person does not open a finance app to read. They open it because something might need doing, and they close it when nothing does. So the app is organised by *time and obligation*: one stream of items, each one a decision with a deadline, an option set and a consequence. Everything else — balances, holdings, history — is reference material that never asks for anything.

	v2 — one answer per screen	v3 — the forward ledger
Organising unit	the screen	the item: one decision, one deadline
Home screen asks	how am I doing?	what needs me, and by when?
Navigation	five topic tabs	two surfaces — Now and Ledger — plus a capture key
Leads with	a figure, then a meaning line	a sentence containing the decision, with the figure inside it
Where actions live	one primary button per screen	on the item, two or three per item, always including *leave it*
Evidence	a disclosure: inputs, rule, engine	a trail: what was seen, in order, plus *what happens if you do nothing*
Success looks like	a screen that reads clearly	an empty queue
Surface style	cards on a canvas, 20dp radii, generous	paper: hairlines, 10dp radii, denser rows, tables for reference
Keeps a record of	figures	figures *and* decisions — what the app said, what you did

> **This is not a correction of v2.** v2's diagnosis of the shipped app was right, and v3 keeps everything v2 got right that is not about structure: the pace bar, the keypad capture, colour never signalling alone, provenance within one tap, demo data, no fixed heights. The disagreement is about what the app is *for*, and that is worth having explicitly.

Five principles

v2 has seven moves about hierarchy. v3 has five about obligation. They are not interchangeable; a screen built to one will fail a review under the other, which is the point of writing both down.

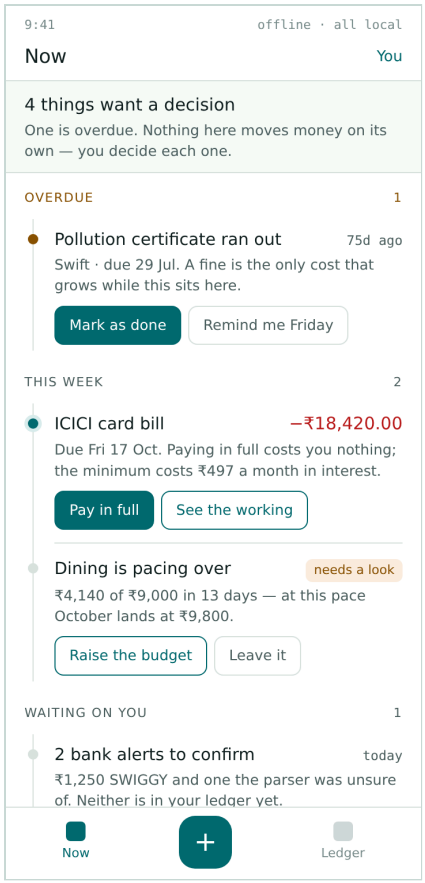
Principle	What it means when drawing a screen
1 · Everything that needs you is one list	Due dates, budget pace, bank-alert drafts, receipts to confirm, the next best rupee, overdue renewals — these are the same kind of thing and belong in one time-ordered stream. If a feature cannot produce an item, ask what it is for.
2 · An item states the decision, not the data	Not “ICICI card · ₹18,420 · due 17 Oct” but “Pay ₹18,420 by Friday, or ₹920 as the minimum — the minimum costs ₹497 a month.” The figure sits inside the sentence that makes it a choice.
3 · Every item carries its own actions, including doing nothing	Two or three chips on the item itself. One of them is always *leave it*, *not now* or *stop flagging this* — an alert you cannot refuse is a nag, and P-07 forbids the app to insist.
4 · The consequence of inaction is stated	Every item answers “what happens if I ignore this?” — because that is the question a queue provokes, and because most of the time the honest answer is *nothing much*, which is information.
5 · The app keeps the record of its own advice	Settled items keep what the app said and what you did. Advice that is never scored is advice nobody can audit, and this app's whole claim is that its figures are checkable.

What is identical to v2 — deliberately

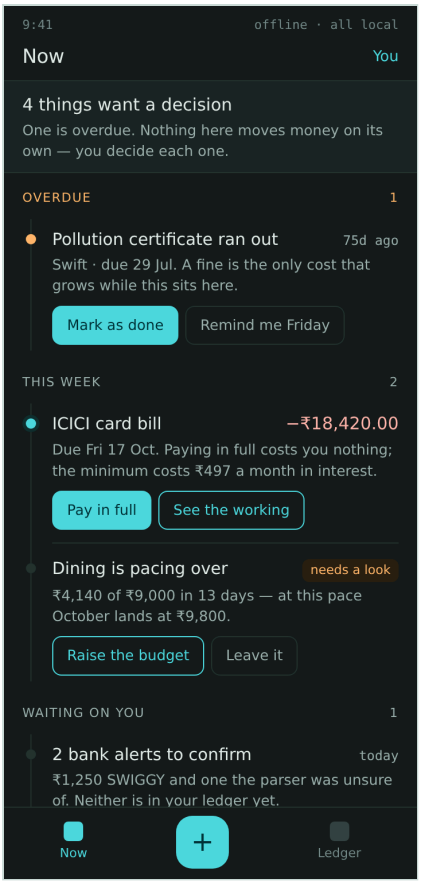
The colour tokens, byte for byte. The seed, the semantic roles, the light and dark values, Roboto, Indian grouping, tabular figures. If these two designs are put in front of users, the variable under test is the structure.

primary #00696E canvas #EEF3F2 surface #FFFFFF surface-2 #F6FAF9 positive #146C2E negative #BA1A1A warning #8A5000

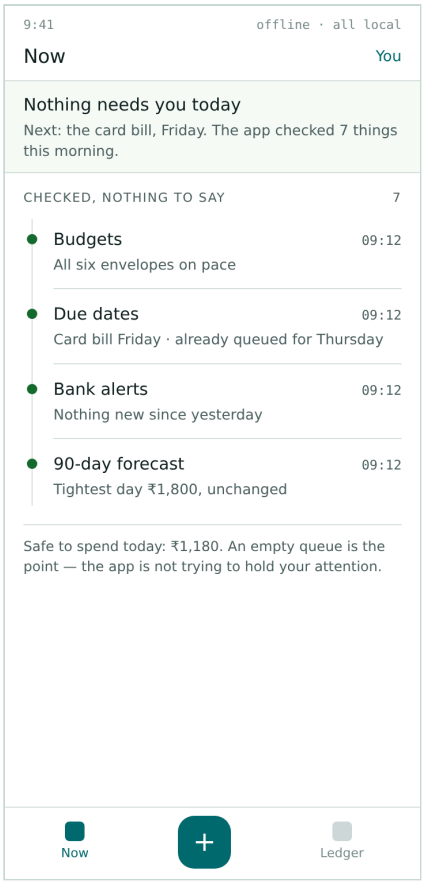
The non-negotiables are unchanged and v3 is held to them exactly as v2 was: colour never signals alone (every flag here carries a word), provenance within one tap (the trail), every string a resource, money through `MoneyFormatter`, no fixed heights, demo data only.



Now — the whole ideology on one screen



The same, dark. Same tokens as v2, and the hairlines still hold



The normal state: nothing to decide, and what was checked

Coverage — every shipped screen, in both designs

The app ships 25 screens (`docs/screens/SCREENS.pdf`). Both designs answer all 25; v3 adds six screens that exist only because of its ideology. Nothing from v1 was dropped on the way to either.

Shipped screen (v1)	v2 mockup	v3 mockup
1. Onboarding	`onboarding`	`onboarding`
2. App lock	`lock`	`lock`
3. Dashboard	`today`	`now · empty`
4. Order of operations	`order`	`order`
5. Add transaction	`add`	`capture`
6. Transactions	`txns`	`txns`
7. Receipt review	`receipt`	`receipt`
8. Bank-alert drafts	`sms`	`item-drafts`
9. Categories	`categories`	`categories`
10. Accounts	`money`	`ledger`
11. Account editor	`accountEditor`	`accountEditor`
12. Holdings	`holdings`	`holdings`
13. Allocation	`allocation`	`allocation`
14. Budgets	`budgets`	`budgets · item-budget`
15. Goals	`goals`	`goals`
16. Goal detail	`goalDetail`	`goalDetail`
17. Emergency fund	`emergency`	`emergency`
18. Purchase advisor	`advisor`	`advisor`
19. Simulators	`simulators`	`simulators`
20. Opportunity	`opportunity`	`opportunity`
21. Vehicles	`vehicles`	`vehicles`
22. Assistant	`ask`	`ask`
23. Settings	`settings`	`you`
24. Consents	`consents`	`consents`
25. Erase everything	`erase`	`erase`
25 screens	**25 designs**	**25 designs**

The six screens that exist only in v3

Screen	Why v2 has no counterpart
item-card	A deadline with options and a consequence. In v2 this is a notification and a row on an account screen; v3 makes it a first-class object.
item-budget	The same, for a condition with no deadline — and the place the *stop flagging this* control lives.
snooze	A queue needs deferral, with the consequence of each option stated. v2 has nothing to defer.
settled	The record of what the app advised and what the user did. v2 keeps figures; v3 keeps decisions.
ahead	The 90-day forecast drawn as a future queue. v2 carries the forecast as a card on Today.
empty	The state most days. A dashboard is never empty, so v2 never has to design this.

> **Read this table as the test of both designs.** An ideology that cannot draw every screen the app already has is a mood board. Both of these can, which is what makes the choice between them a real one.

The ladder — v1 to v3, screen by screen

v2 was argued against the shipped app. v3 has to be argued against **both** — against what ships today and against v2 — or it is only a style. Every screen section that follows shows all three, in order, and states what changed at each step.

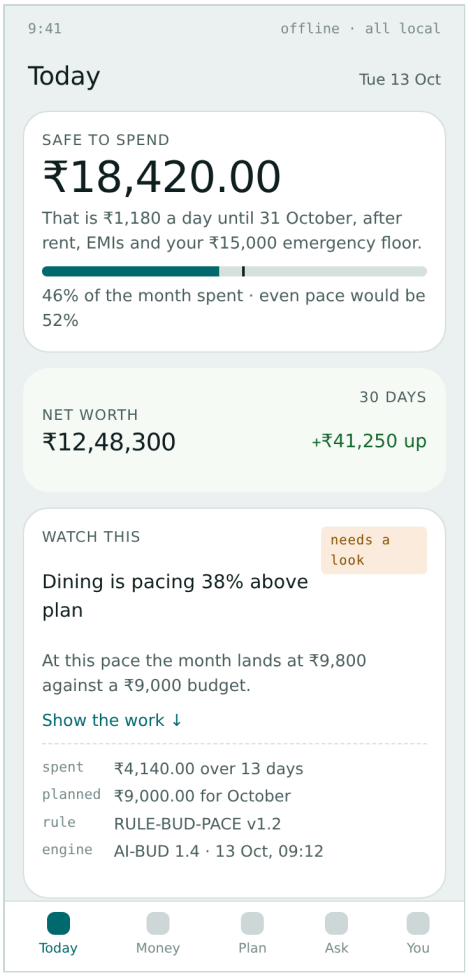
Shipped screen	What v2 changed from v1	What v3 changes from v2
Onboarding	the pledge becomes a preview of the queue the user will actually meet	three questions instead of four steps, and the empty state is part of the pitch
App lock	the PIN text field becomes dots and a keypad that cannot be covered by the keyboard	same fix, in v3's chrome: the tries-left warning sits in the counter line
Dashboard	the fourteen navigation rows and the six derivation lines are gone	the figure stops leading: what needs a decision does
Order of operations	eight stacked cards become a rail with a state and an amount per step	the same list idiom as everything else, rather than a shape used once
Add transaction	amount-first with its own keypad; no system keyboard, no ragged buttons	identical in substance; capture is a key in the bar rather than a tab
Transactions	day totals become structure and the per-row delete icon is gone	a table rather than cards, because the ledger is reference material
Receipt review	the duplicate warning rises above the fields and merge becomes the primary	the duplicate becomes an item with its own two actions
Bank-alert drafts	drafts get per-draft actions and the unsure one says why	drafts join the single queue instead of living on their own screen
Categories	nature becomes visible on every row	adds transaction counts, so the taxonomy can be audited
Accounts	the four buttons per row are gone and the rows group into own/owe	tables rather than cards, and the surface states that it never asks for a decision
Account editor	the EMI the terms produce leads the screen	adds where this account shows up next — a dated claim, a step in the order
Holdings	price age moves into the chrome and an unpriced holding says so	the §11.1 disclaimer moves into the trail rather than sitting as a banner
Allocation	the split is drawn and the flag carries its rule	a bar and a table instead of a donut, and the cap becomes editable
Budgets	pace is drawn once per envelope instead of described three times	states its relationship to the queue: the alert lives on Now, this is where you plan
Goals	ten lines of prose per goal become a row, a bar and a pill	goals are claims in priority order, not progress cards
Goal detail	the evidenced/claimed split is drawn and named	the gap becomes a decision with three ways to close it
Emergency fund	months of cover lead instead of the target amount	the coach line becomes an item that names what it displaces
Purchase advisor	what the purchase leaves becomes the figure; the verdict is a chip	adds the record: whatever you choose goes into Settled
Simulators	both sides and the gap lead, with the flip rate stated	a table rather than two cards, and the assumptions become chips
Opportunity	the score is out of what was possible and the gates are visible	adds a reminder action — leave with a condition rather than a feeling
Vehicles	the prediction carries its basis and overdue is a word with a number	the overdue renewal is the same object as the item on Now
Assistant	a conversation layout with the composer pinned and the trace on the answer	an answer can become an item, and only the user files it

Settings	grouped by what each control changes, with the backup warning where it is acted on	tables, with every state visible without opening a row
Consents	each row gains what stops if it is withdrawn, beside the switch	each consent becomes an item with its action inline
Erase everything	the scope is counted and the caveat is the loudest thing on the screen	the count includes the settled decisions v3 itself created

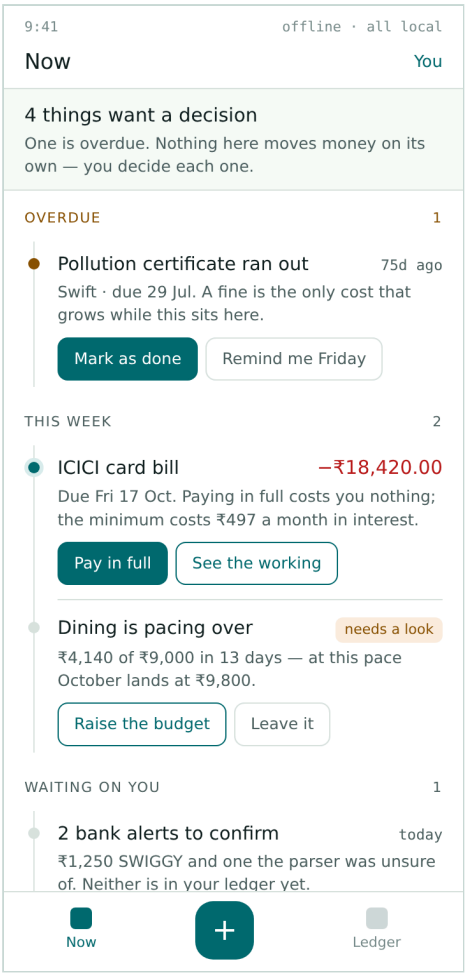
> **Where the two proposals agree, that is worth noticing.** The lock screen, the capture keypad and the pace bar come out the same in both — which is the strongest evidence any of it is right, since the two designs disagree about nearly everything else.

Head to head · The home screen

The same job, done by two ideologies. Both are drawn in the same palette.



v2 — Today



v3 — Now

What v2 does better

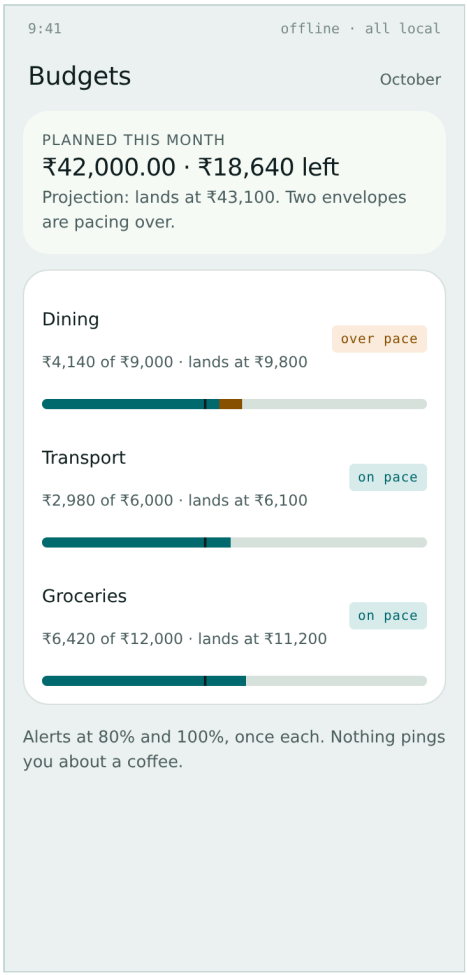
- Answers *how am I doing* in one glance: one figure, one sentence, one pace bar. A user who opens the app to check a number is finished in two seconds.
- Calmer. Four cards of air against nine dense rows — and for a daily-use finance app, calm is a feature, not a shortfall.
- Nothing on it can be mistaken for an instruction, which matters for P-07 on the screen people see most.

What v3 does better

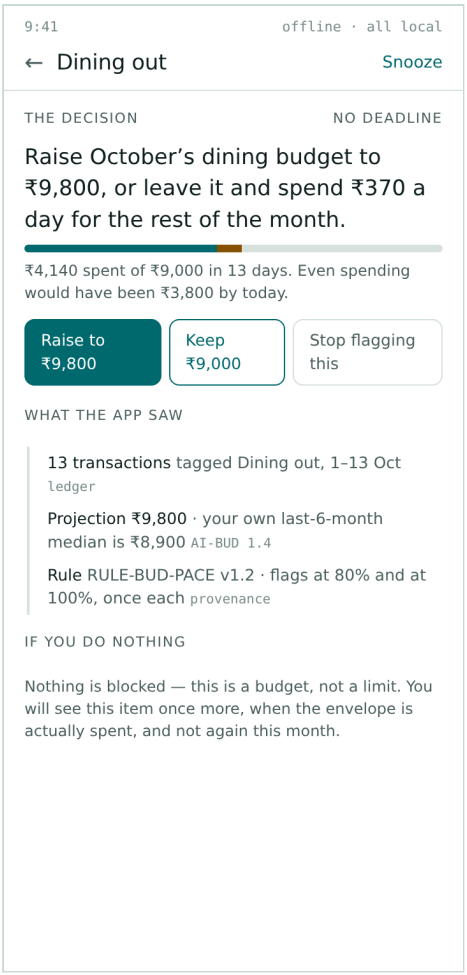
- Answers *what needs me*, which is the question that actually makes someone open the app on a Tuesday.
- The overdue certificate and the card due Friday are above the fold. On v2's Today they are an insight card and a notification respectively — easy to scroll past.
- You can act without navigating: four decisions are one tap each from the first screen.
- It can be empty, and says what it checked. A dashboard is never empty, so it never tells you that everything is fine.

Head to head · A budget going over

The same job, done by two ideologies. Both are drawn in the same palette.



v2 — the Budgets screen



v3 — the item it produces

What v2 does better

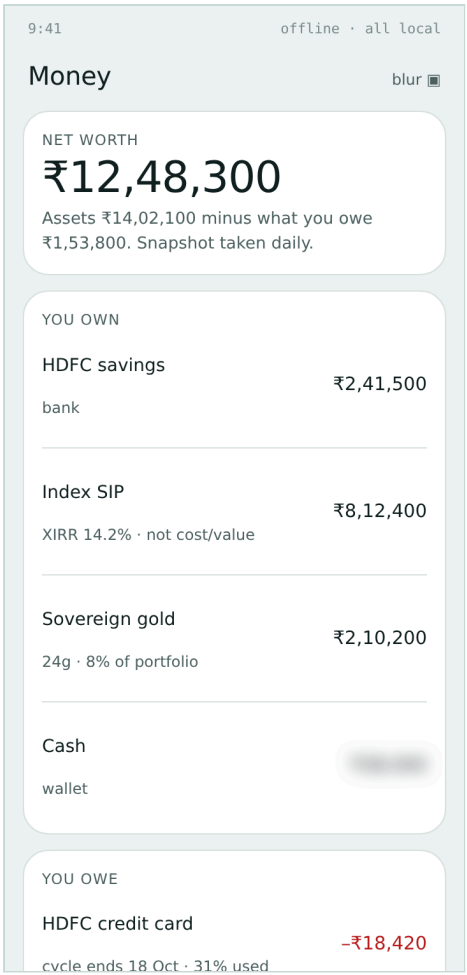
- Shows all six envelopes at once: the shape of the month, not just the one that broke.
- Better for the user who budgets deliberately — planning is comparison, and v3 hides the other five.
- One screen to learn, reachable any time; v3's item exists only while something is wrong.

What v3 does better

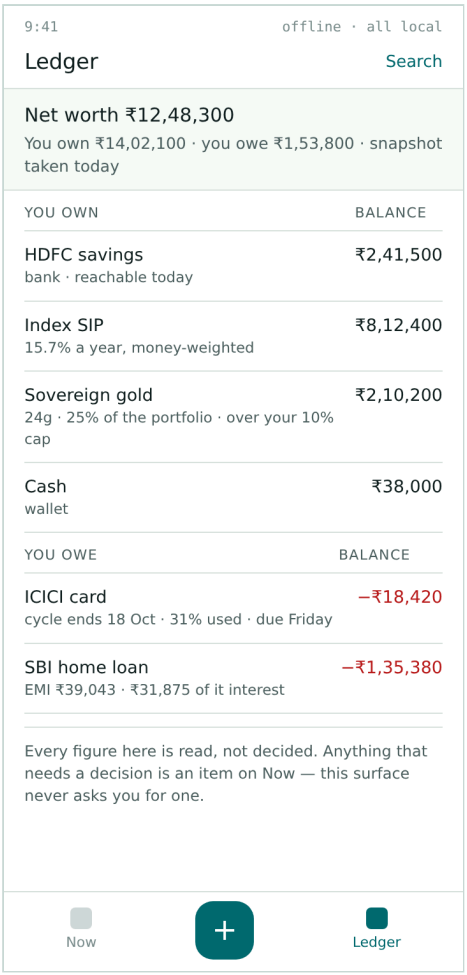
- Frames it as a choice with two named outcomes — raise to ₹9,800, or spend ₹370 a day — where v2 states the pace and leaves the user to work out what to do about it.
- Carries *stop flagging this*, so an alert the user disagrees with can be ended at the point of annoyance (P-07).
- Says what happens if nothing is done: nothing is blocked, and it will not ask again this month. That is what stops a queue from becoming a nag.

Head to head · Accounts and balances

The same job, done by two ideologies. Both are drawn in the same palette.



v2 — Money



v3 — Ledger

What v2 does better

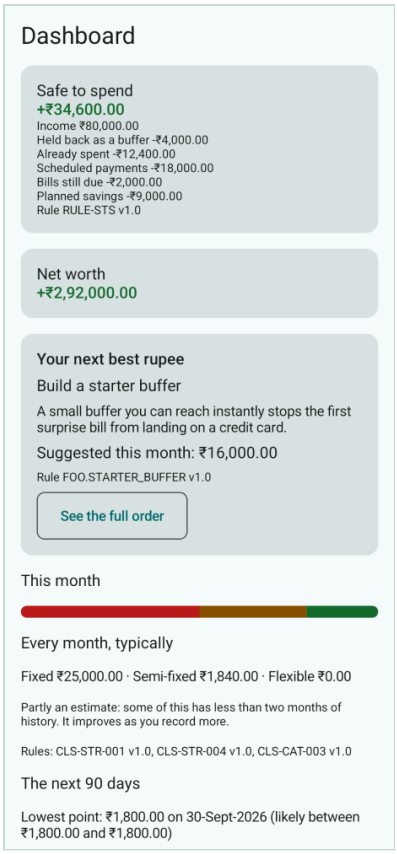
- Cards give each account room: a credit card's cycle and utilisation read as a small story rather than a sub-line.
- Net worth leads with the meaning line that explains it, which is the better teaching surface.
- Scales down gracefully to a user with two accounts; a table of two rows looks empty.

What v3 does better

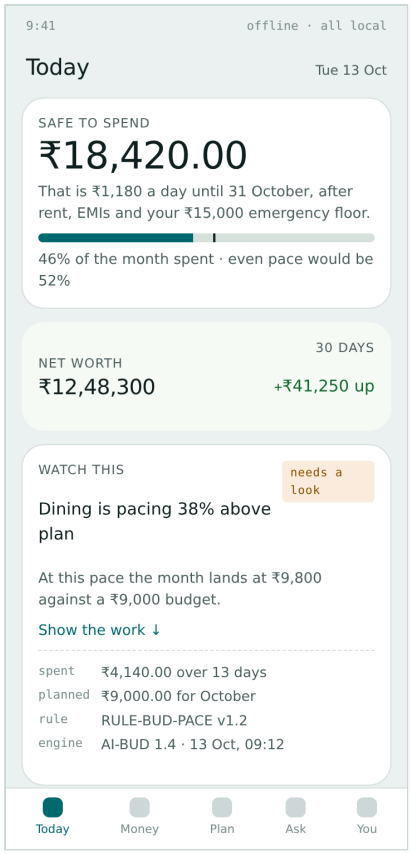
- Denser: eight accounts fit where v2 fits five, and a table is what people expect to scan for a balance.
- States its own role — *this surface never asks you for a decision* — which is only possible because another surface does.
- Scales up better: twenty accounts stay readable, where twenty cards become a scroll.

1 · Now — the stream

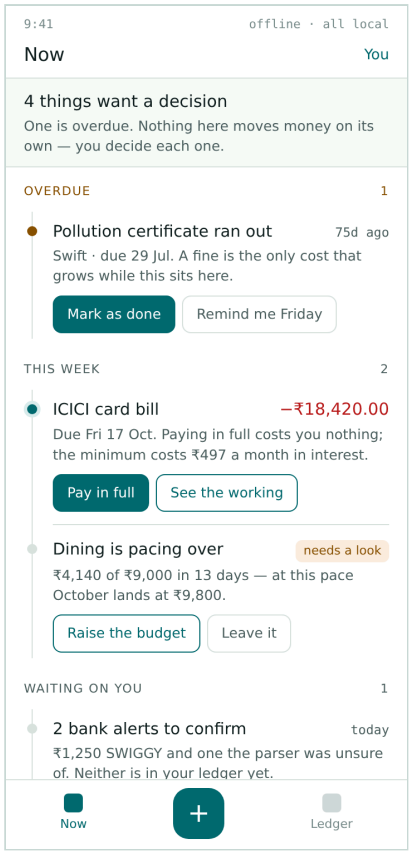
the home surface · replaces the dashboard · principles 1, 2, 3



v1 — ships today



v2 — one answer per screen



v3 — the forward ledger

> **v3 vs v1:** the fourteen navigation rows and the six derivation lines are gone
>
> **v3 vs v2:** the figure stops leading: what needs a decision does

The idea

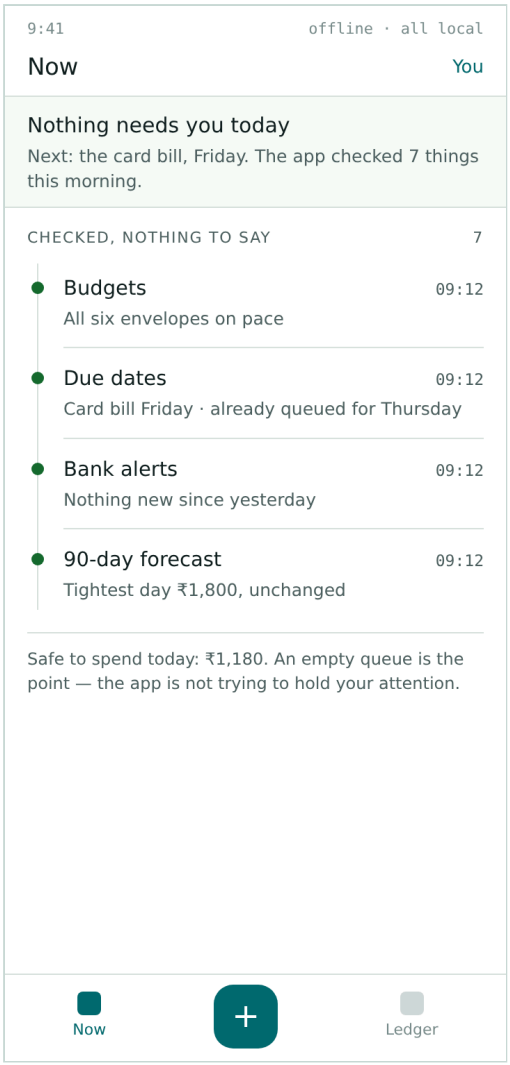
- Three time buckets — **Overdue**, **This week**, **Waiting on you** — down a single rail, newest obligation first.
- A counter at the top states the size of the queue in words and promises that nothing moves on its own.
- Each item is a sentence, a figure, and its own two or three actions.
- Safe-to-Spend survives as a footer line, not a headline.

Why, and what it costs

- The engines already produce a queue — budget alerts, due dates, drafts, renewals, the next best rupee. v2 scatters those across five tabs; v3 admits they are one list. That is a structural observation about this codebase, not a style preference.
- Actions on the item mean a decision costs one tap from the first screen, and the thumb never leaves the lower half (Fitts, and Material 3's own guidance for list actions).
- Buckets are time, not topic, because every one of these items has a deadline or is explicitly marked as having none.
- Demoting Safe-to-Spend is the cost of the ideology, and it is a real cost — see the head-to-head above.

2 · Now, empty

the state most days · principle 5



v3 — Now, empty

> **No counterpart in v1 or v2.** This screen exists because of v3's premise; there is nothing to compare it against, which is the honest way to show what an ideology adds.

The idea

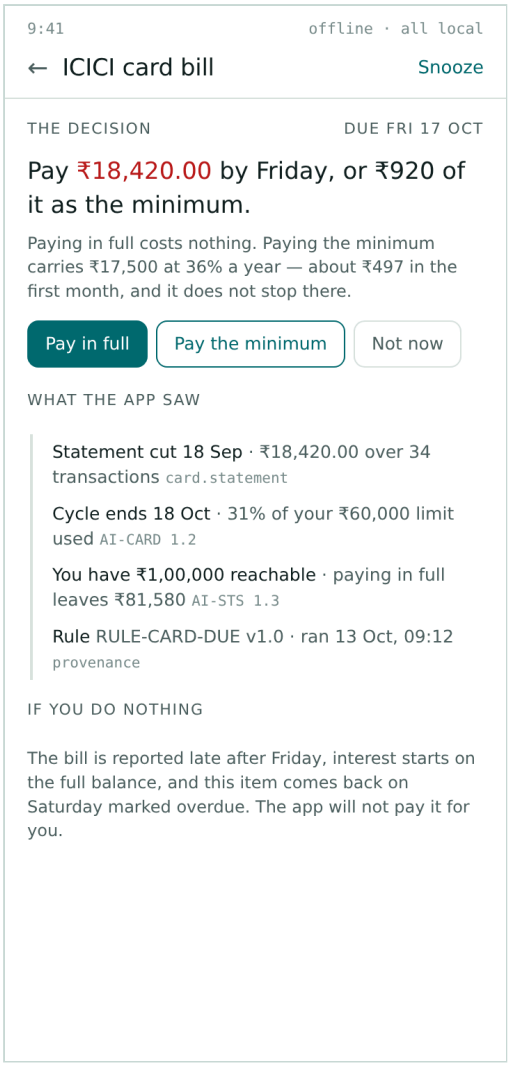
- "Nothing needs you today", the next thing that will, and a list of what the app checked this morning with the time it ran.
- The checked items are shown settled, not hidden.

Why, and what it costs

- An inbox that only looks right when it is full teaches anxiety. The empty state is the one v3 must get right, because it is the common case for a user whose finances are in order.
- Showing the seven checks that found nothing is what makes the empty screen trustworthy rather than suspicious — it is the difference between "nothing to report" and "nothing was looked at".
- It also says, in effect, that the app is not trying to hold attention — which is the only honest position for software that runs entirely on the user's phone and sells nothing.

3 · An item with a deadline

the card bill · principles 2, 4



v3 — An item with a deadline

> **No counterpart in v1 or v2.** This screen exists because of v3's premise; there is nothing to compare it against, which is the honest way to show what an ideology adds.

The idea

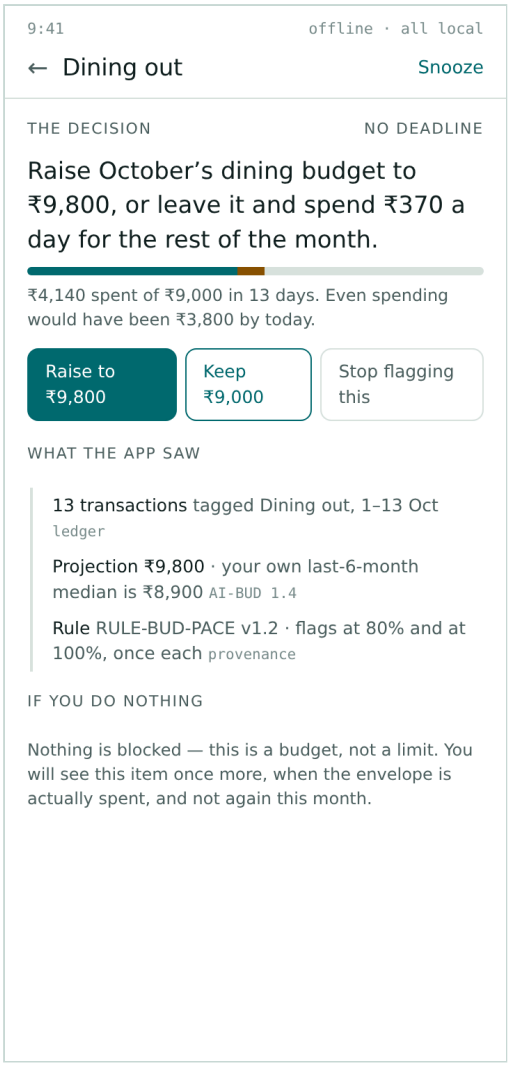
- The decision is the first line, with both options and the price of the cheaper-looking one.
- The trail lists what the app saw, in the order it saw it, each line carrying its source.
- **If you do nothing** is a section, not a footnote.

Why, and what it costs

- ₹497 is the number that decides this, and on a conventional bill screen it does not appear at all. An item exists to carry the comparison, not the balance.
- The trail is v2's disclosure turned chronological: inputs, the engine's reading, the rule and when it ran. Same obligation (P-02, AI-ARC-003), different shape — a queue item is read once, so the evidence is open by default rather than behind a tap.
- Stating the consequence is what lets a user ignore an item honestly. Without it, every item implies urgency and the queue trains dismissal.

4 · An item with no deadline

budget pace · principles 3, 4



v3 — An item with no deadline

> **No counterpart in v1 or v2.** This screen exists because of v3’s premise; there is nothing to compare it against, which is the honest way to show what an ideology adds.

The idea

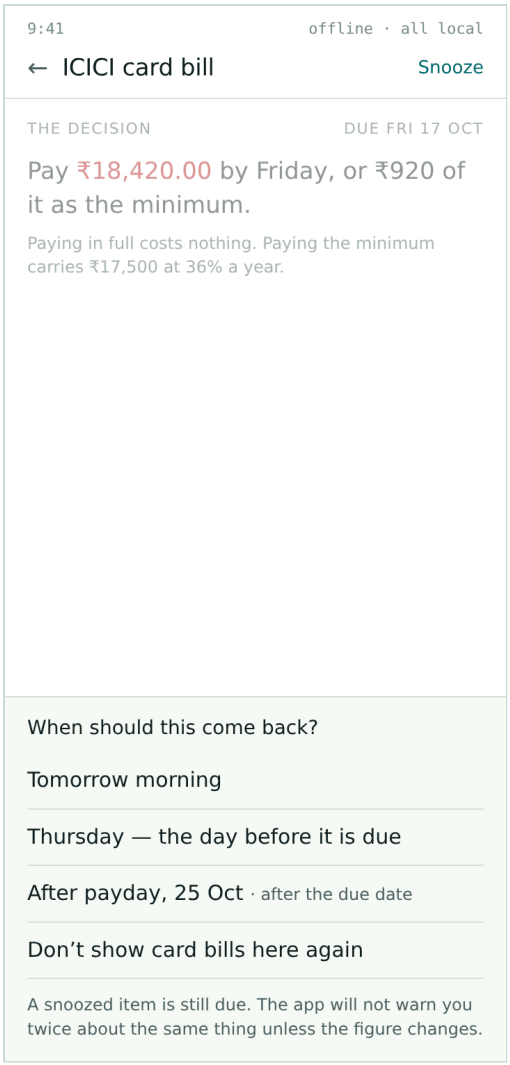
- Two named outcomes, the pace bar v2 designed, and a third chip: **stop flagging this**.
- The projection stays in words; the bar carries spent, pace and overshoot.

Why, and what it costs

- v2's move 4 was right and v3 keeps it unchanged. A different ideology is not a licence to redo what works.
- The mute chip is the ideological difference: in a queue, the user must be able to end a conversation. An alert that cannot be refused is an order, and P-07 says the app does not give those.
- “Nothing is blocked — this is a budget, not a limit” is the sentence that keeps a finance queue from feeling like a debt collector.

5 · Snoozing

the queue's core interaction · principle 3



v3 — Snoozing

> **No counterpart in v1 or v2.** This screen exists because of v3’s premise; there is nothing to compare it against, which is the honest way to show what an ideology adds.

The idea

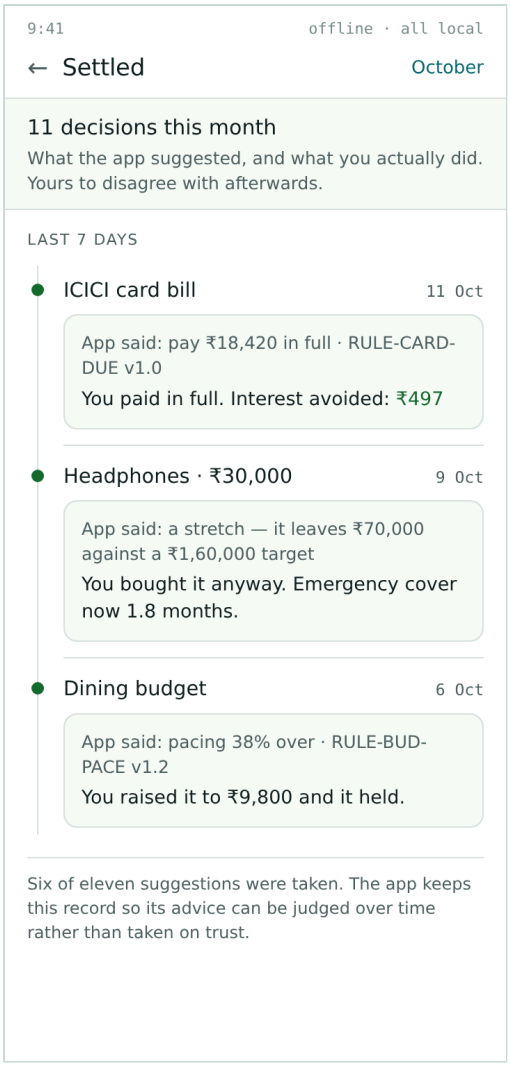
- Four options, each stating its consequence — including *after payday, 25 Oct*, labelled **after the due date**.
- “Don't show card bills here again” is one of the four.
- A line states that a snoozed item is still due, and that the app will not warn twice about the same figure.

Why, and what it costs

- Any queue lives or dies on its snooze. If deferring is hard, users dismiss instead, and dismissal loses the obligation entirely.
- Labelling the option that lands after the deadline is the difference between a helpful default and a trap (Nielsen #5, error prevention).
- The no-repeat rule is the notification policy (§17.2, AI-NTF) made visible: the app states its own restraint where the user can hold it to it.

6 · Settled — the record of advice

new in v3 · principle 5



v3 — Settled — the record of advice

> **No counterpart in v1 or v2.** This screen exists because of v3's premise; there is nothing to compare it against, which is the honest way to show what an ideology adds.

The idea

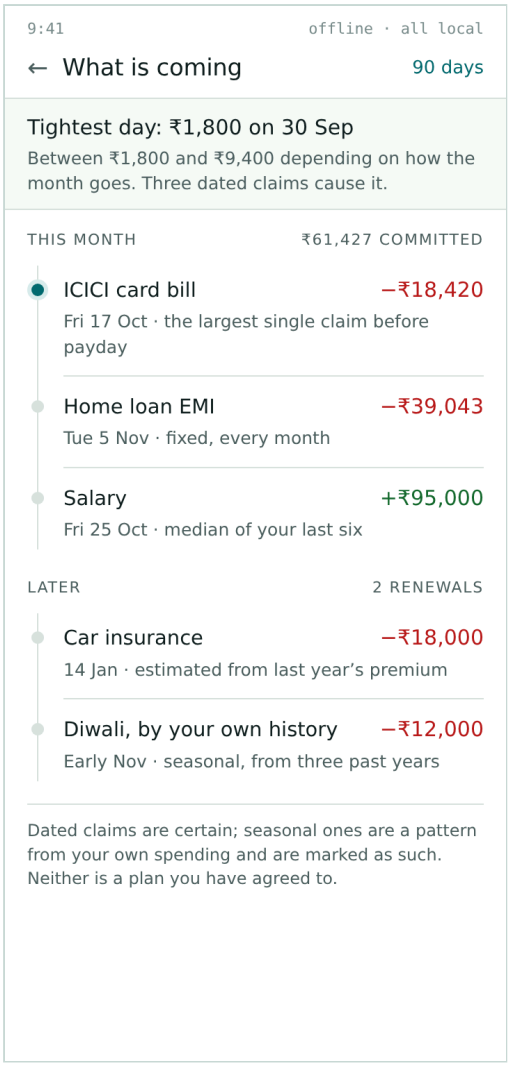
- Every decided item keeps two lines: **App said**, with the rule that produced it, and **You did**, with what followed.
- A counter: six of eleven suggestions taken this month.

Why, and what it costs

- This is v3's strongest claim and its biggest ask. An app whose whole case is "every figure is checkable" should also be able to show whether its advice was any good — and whether the user was right to ignore it.
- It is also the honest counterpart of P-07: if the user overrules the app and nothing bad happens, the record says so, and the app has to live with that.
- **It is the one thing here that needs new storage.** The app has an `audit_log` for security events, not a decision log. That is a schema change and an ADR, and it is the main reason v3 costs more than v2 — see the closing section.

7 · What is coming

the 90-day forecast as a future queue · principles 1, 4



v3 — What is coming

> **No counterpart in v1 or v2.** This screen exists because of v3's premise; there is nothing to compare it against, which is the honest way to show what an ideology adds.

The idea

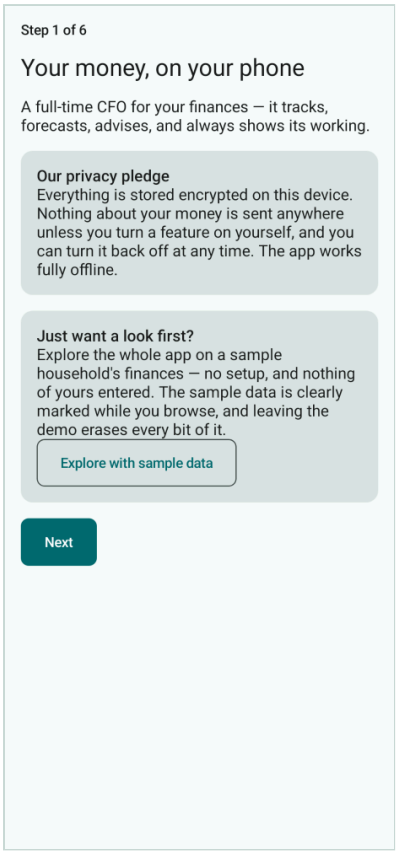
- The tightest day is the counter; the claims that cause it are items on the same rail.
- Dated claims and seasonal patterns are drawn the same way but labelled differently.

Why, and what it costs

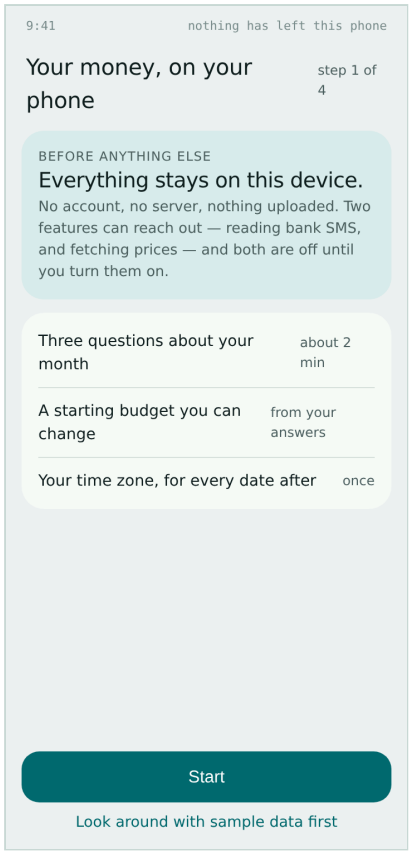
- A forecast is a queue that has not arrived yet, so it uses the same mental model and the same component. The user learns one pattern.
- Marking which claims are certain and which are inferred from their own history is P-03 in a layout: the app may not present a pattern as a commitment.
- "Neither is a plan you have agreed to" keeps the forward view advisory (P-07), which matters more here than anywhere, because a list of future payments reads like a schedule.

8 · Onboarding

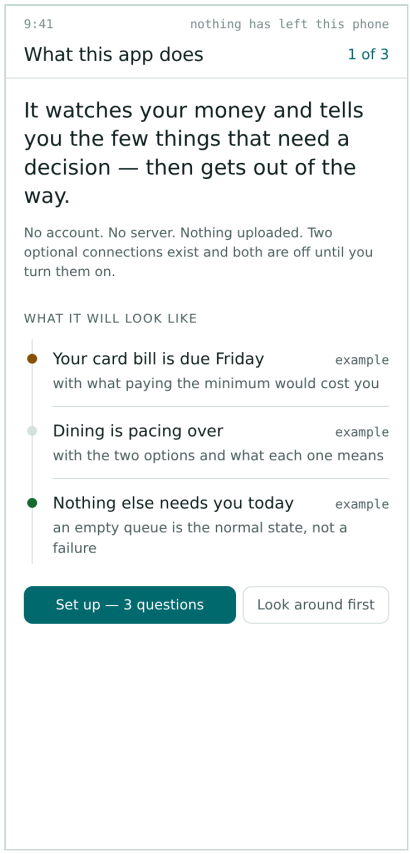
the offer · principle 2



v1 — ships today



v2 — one answer per screen



v3 — the forward ledger

> **v3 vs v1:** the pledge becomes a preview of the queue the user will actually meet
>
> **v3 vs v2:** three questions instead of four steps, and the empty state is part of the pitch

The idea

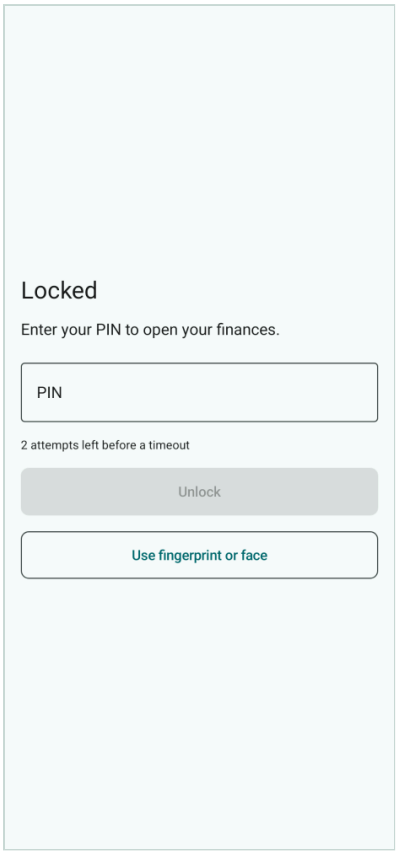
- One sentence of what the app does, then three example items — including the empty one.
- The privacy position is the second paragraph, not a card.

Why, and what it costs

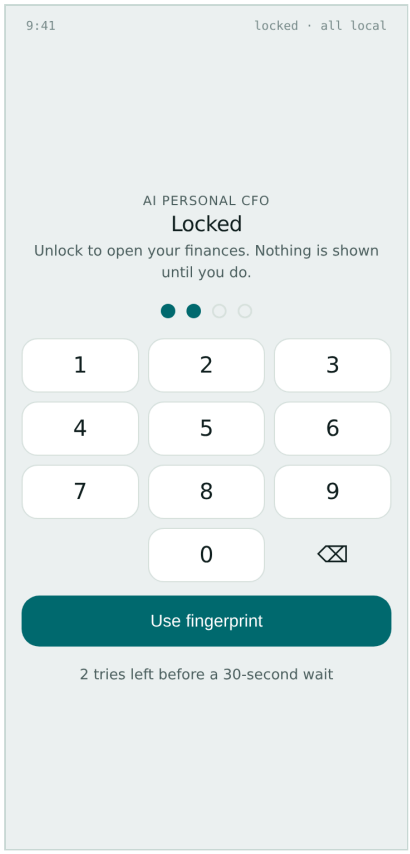
- If the app is a queue, onboarding should show the queue. Promising “we will tell you the few things that need a decision, then get out of the way” is a claim the first screen can be held to.
- Showing the empty item sets the expectation that most days there is nothing — which is the honest pitch and the one that survives contact with a well-run household.
- Three questions rather than v2's four steps, because v3 needs less to start: the queue fills from the ledger as transactions arrive.

9 · App lock

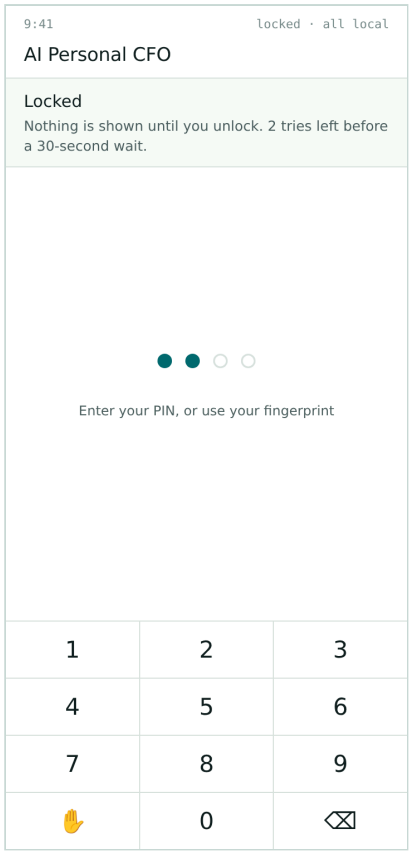
SEC-002, §23.1 · principle 2



v1 — ships today



v2 — one answer per screen



v3 — the forward ledger

> **v3 vs v1:** the PIN text field becomes dots and a keypad that cannot be covered by the keyboard
> **v3 vs v2:** same fix, in v3's chrome: the tries-left warning sits in the counter line

The idea

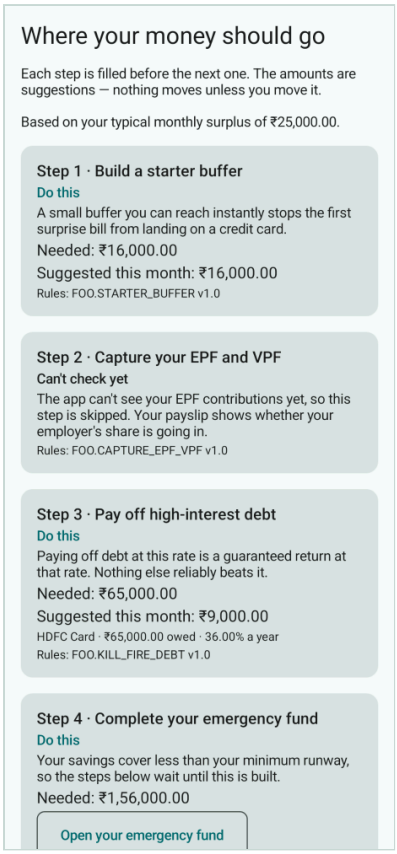
- Counter line states the lock and the tries left; four dots; a full-bleed keypad.
- Fingerprint is a key on the pad rather than a separate button.

Why, and what it costs

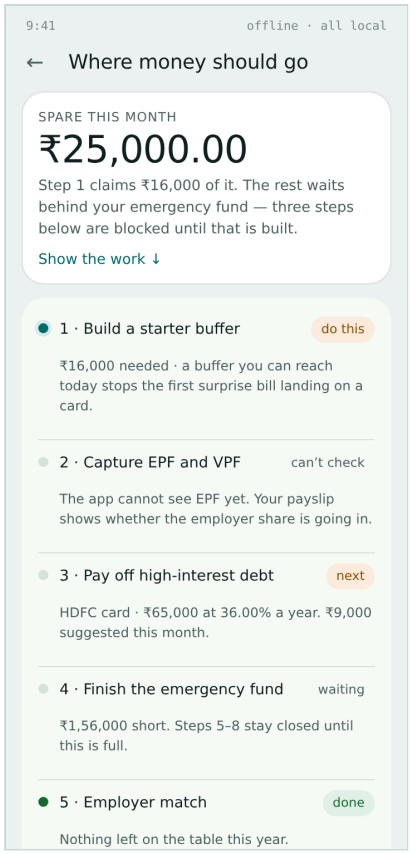
- Same reasoning as v2 — a text field summons the keyboard that hides the action at 200% font — reached through v3's denser chrome, so the keypad fills the screen rather than sitting in a card.
- The counter carries “2 tries left” where v3 always puts status, so the warning is in the same place as on every other screen.

10 · Where the spare goes

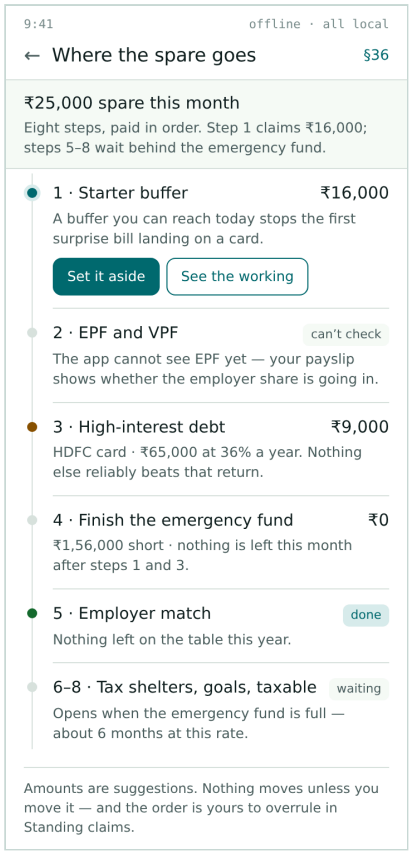
§36, FOO-002 · principles 1, 2



v1 — ships today



v2 — one answer per screen



v3 — the forward ledger

> **v3 vs v1:** eight stacked cards become a rail with a state and an amount per step
>
> **v3 vs v2:** the same list idiom as everything else, rather than a shape used once

The idea

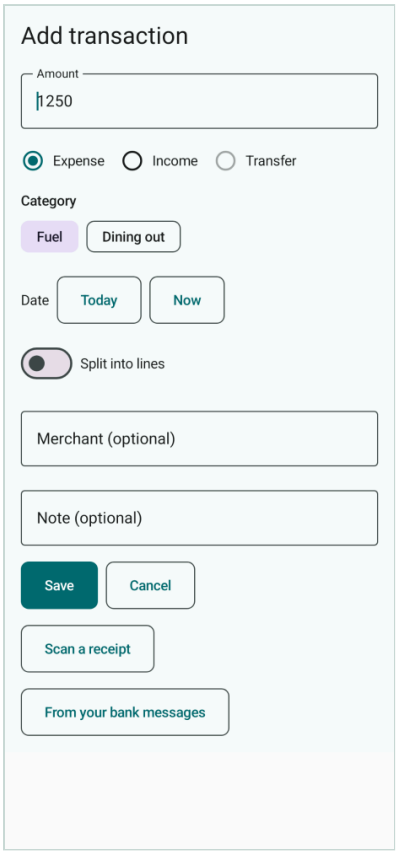
- The eight stages are items on the same rail as everything else, with amounts and status flags.
- The first stage carries actions; the blocked ones say what opens them and when.

Why, and what it costs

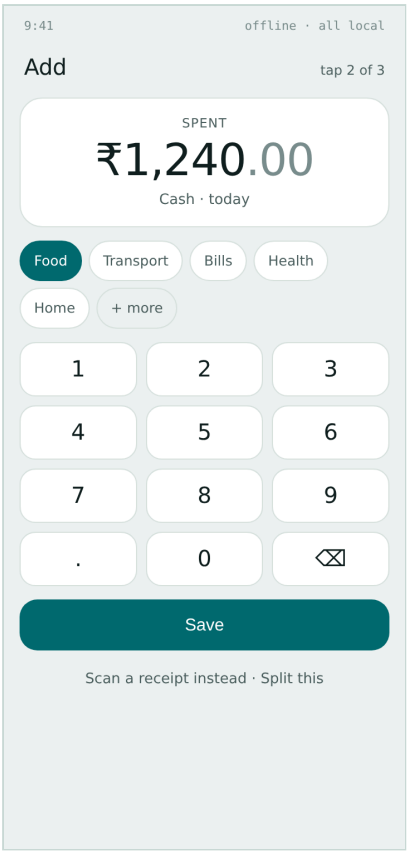
- In v3 this is not a separate mental model — it is the queue, filtered to one question. The user learns one list idiom and sees it again here.
- “Opens when the emergency fund is full — about 6 months at this rate” turns a blocked row into a date, which is what makes a waterfall bearable.

11 · Capture

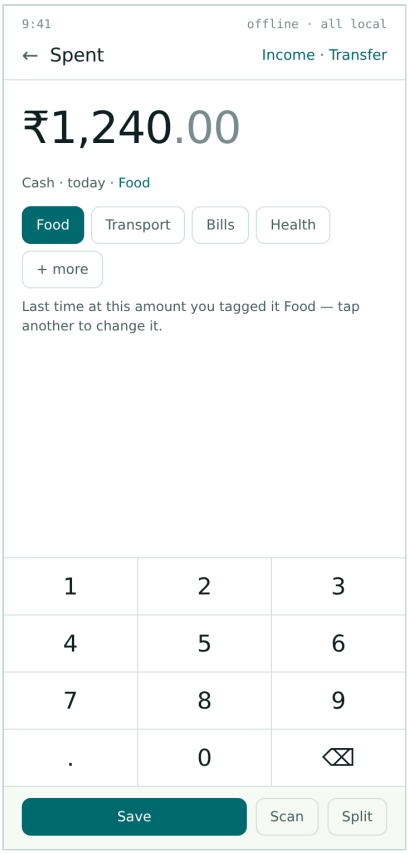
the bottom-bar key · unchanged from v2 in substance



v1 — ships today



v2 — one answer per screen



v3 — the forward ledger

> **v3 vs v1:** amount-first with its own keypad; no system keyboard, no ragged buttons
>
> **v3 vs v2:** identical in substance; capture is a key in the bar rather than a tab

The idea

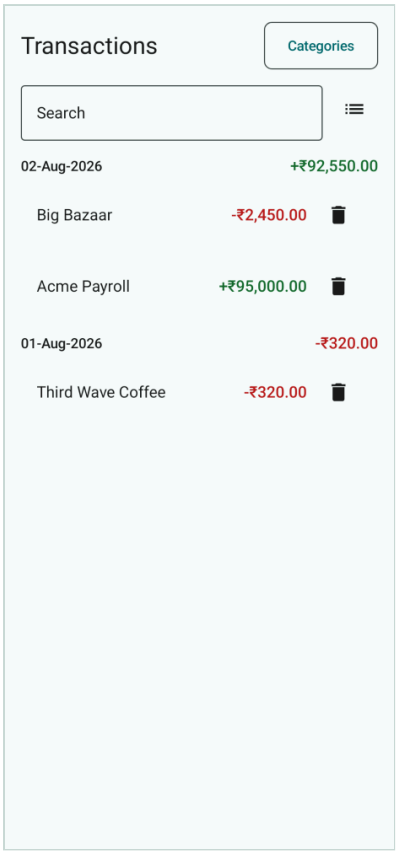
- Amount first, own keypad, category chips, one Save.
- A line recalls what this amount was tagged last time.
- It is a key in the bottom bar rather than a tab.

Why, and what it costs

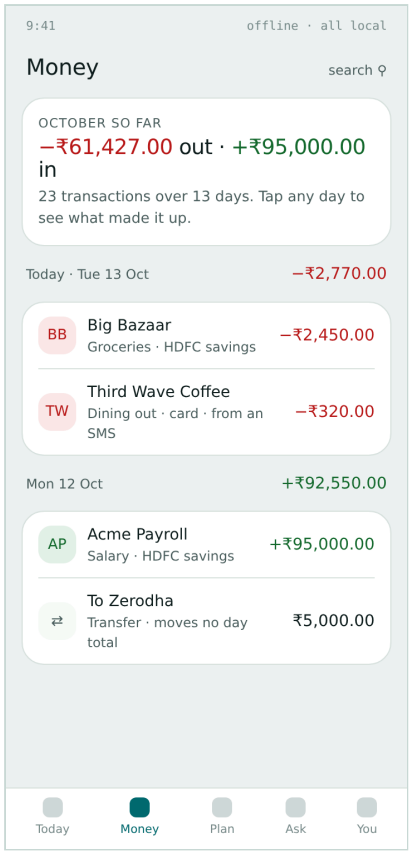
- v2 solved this screen; copying it is the right move. FR-TXN-002's three-tap budget is met the same way, and the two designs would produce the same Paparazzi baseline here.
- The one change is placement: capture is an action, not a destination, so it is a key and not a tab — which also keeps the bottom bar at two surfaces.
- The recall line is a small concession to speed that costs nothing: the suggestion is the classifier's existing output (§8.1), shown where it saves a tap.

12 · Transactions

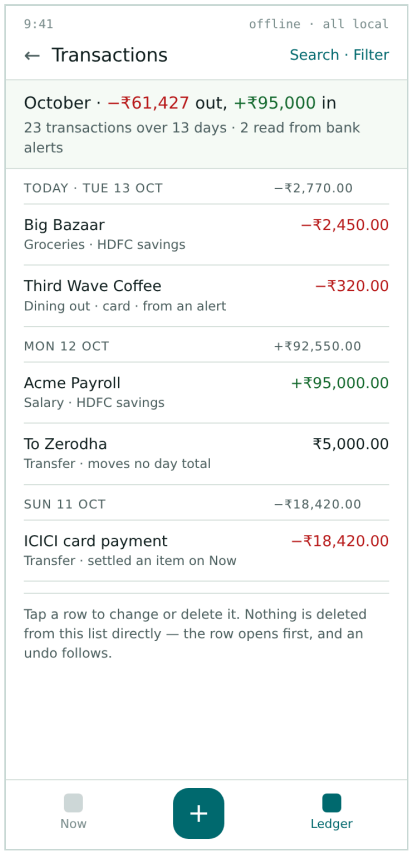
FR-TXN-007 · principle 1



v1 — ships today



v2 — one answer per screen



v3 — the forward ledger

> **v3 vs v1:** day totals become structure and the per-row delete icon is gone
>
> **v3 vs v2:** a table rather than cards, because the ledger is reference material

The idea

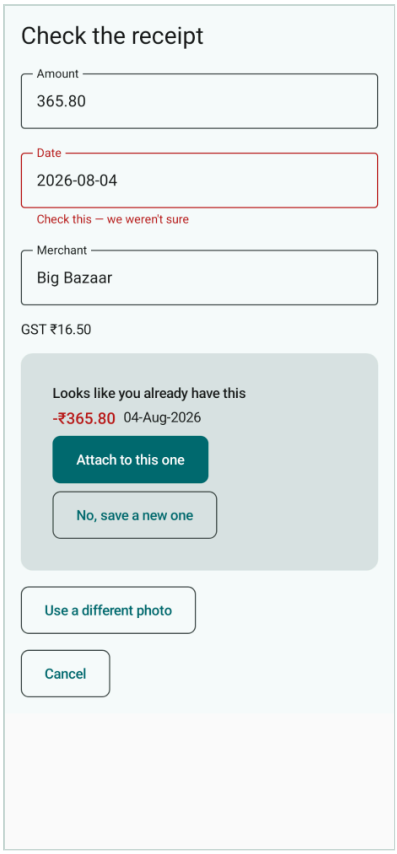
- A table under Ledger, grouped by day with the day's net in the header row.
- The counter answers the month: out, in, and how many of them came from bank alerts.
- A row opens before it can be deleted; the undo is unchanged.

Why, and what it costs

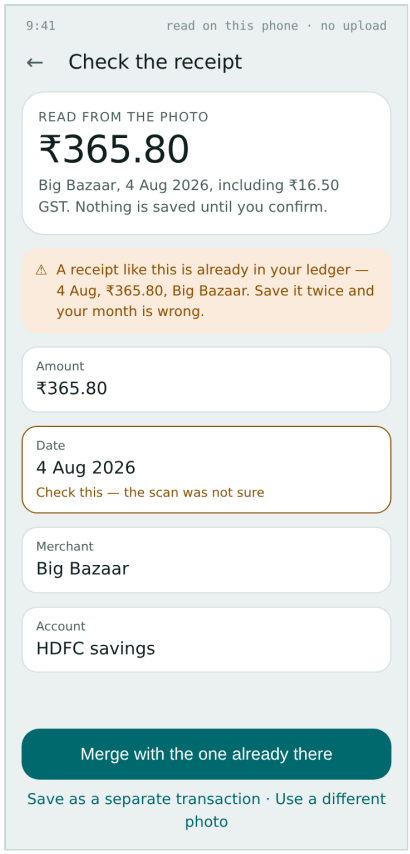
- Tables are what people scan for a transaction, and v3 is already a table-led surface — a card per row would cost twice the height for the same four facts.
- The ledger never asks for a decision; anything that needs one is an item on Now. That separation is what keeps this list calm enough to be long.

13 · Receipt review

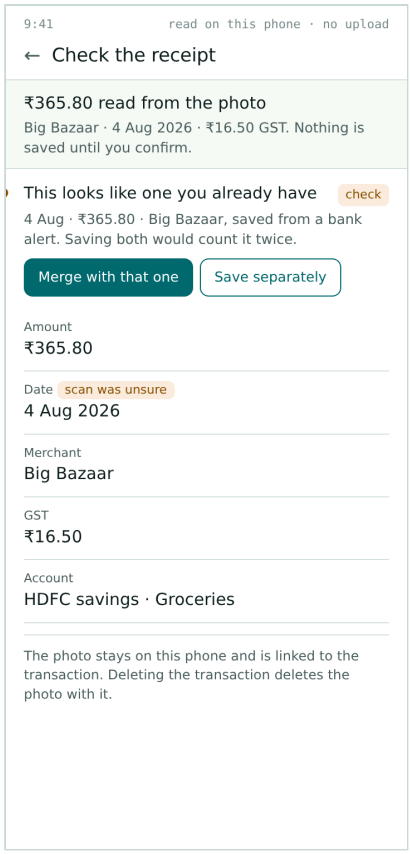
FR-OCR-001…006 · principles 2, 4



v1 — ships today



v2 — one answer per screen



v3 — the forward ledger

> **v3 vs v1:** the duplicate warning rises above the fields and merge becomes the primary
>
> **v3 vs v2:** the duplicate becomes an item with its own two actions

The idea

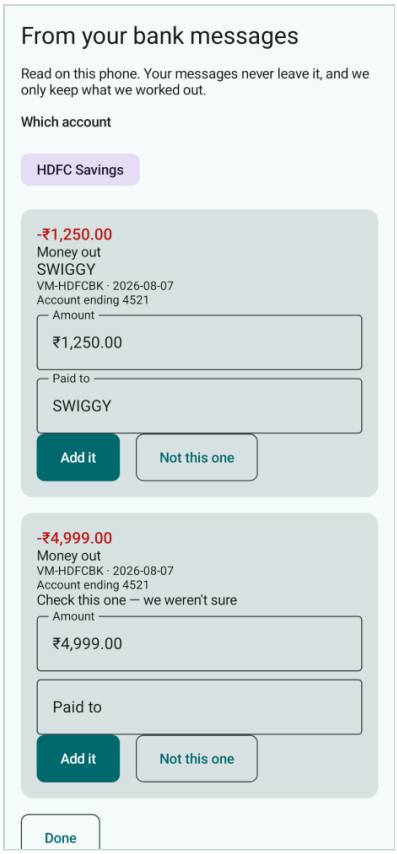
- The read total is the counter; the duplicate match is an item with its own two actions.
- The uncertain field carries a flag beside its label rather than a coloured outline alone.

Why, and what it costs

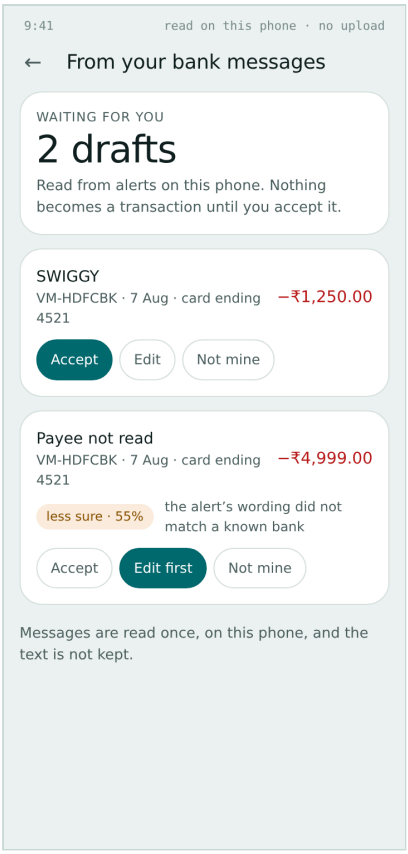
- The duplicate is a decision, so it gets the item treatment the rest of the app uses for decisions — merge or save separately, stated as outcomes.
- The closing line answers the question a photo raises and no screen in v1 answered: where the image goes, and what deleting the transaction does to it.

14 · Confirming what was read

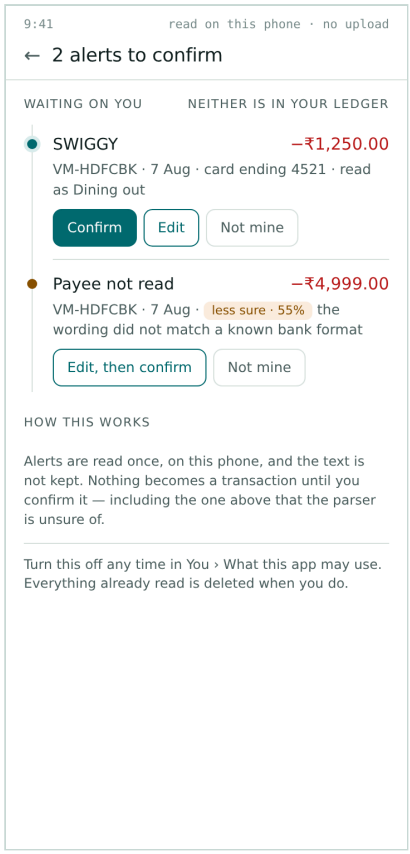
bank-alert drafts · §18, §23 · principles 1, 3



v1 — ships today



v2 — one answer per screen



v3 — the forward ledger

> **v3 vs v1:** drafts get per-draft actions and the unsure one says why
>
> **v3 vs v2:** drafts join the single queue instead of living on their own screen

The idea

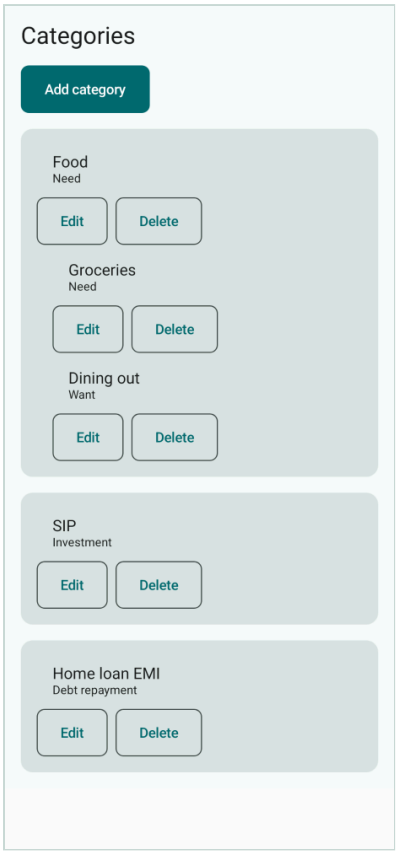
- Drafts are items on the same rail as everything else, with per-draft confirm, edit and *not mine*.
- The unsure draft states its confidence and what the parser could not read.
- The consent story sits at the foot: read once, not kept, revocable in one place.

Why, and what it costs

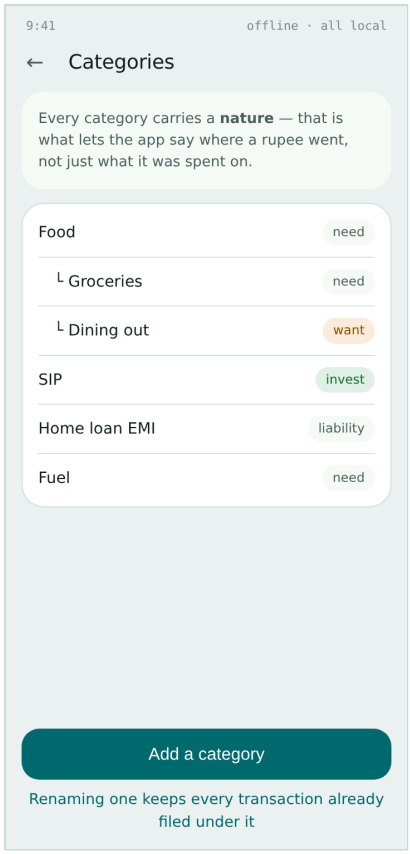
- This is the clearest case for principle 1: a draft is exactly a thing waiting on the user, and in the shipped app it lives behind two taps on another screen where it is forgotten.
- Per-draft actions, never a bulk accept — the review is the feature (P-07).
- Keeping the consent sentence visible where the data appears, rather than only on the privacy screen, is what makes P-01 legible at the point it matters.

15 · Categories

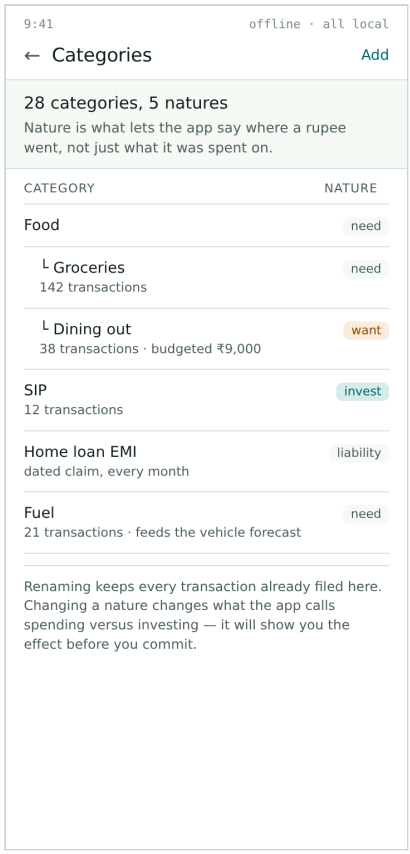
FR-SET-001, §8.3 · principle 1



v1 — ships today



v2 — one answer per screen



v3 — the forward ledger

> **v3 vs v1:** nature becomes visible on every row
>
> **v3 vs v2:** adds transaction counts, so the taxonomy can be audited

The idea

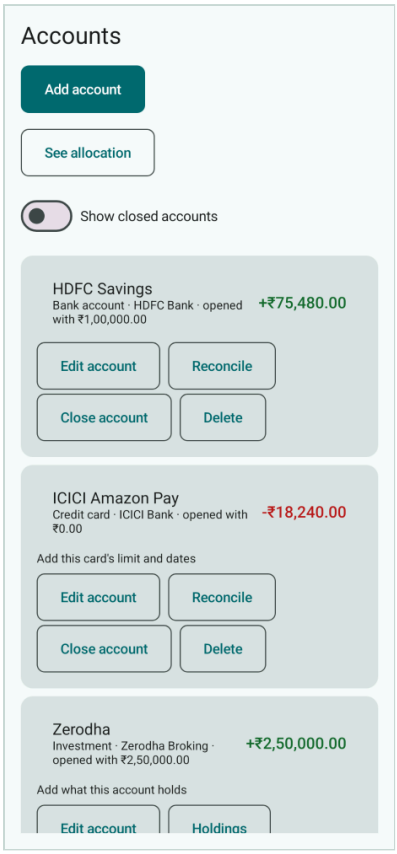
- A table with the nature in the right-hand column and the transaction count under each name.
- The closing line says what a rename keeps, and what changing a nature changes.

Why, and what it costs

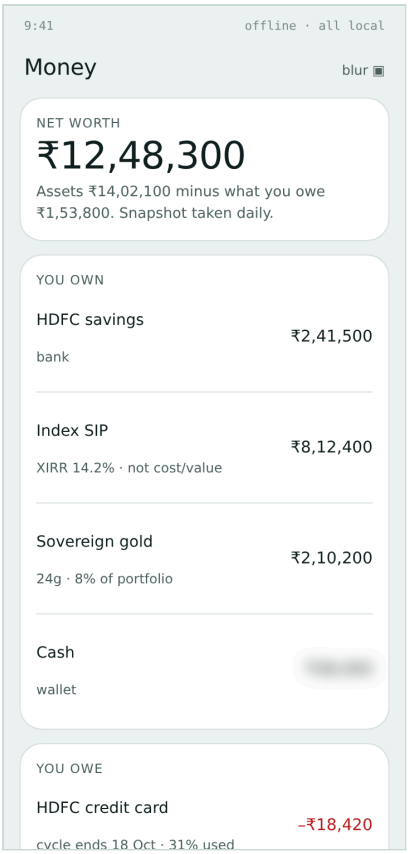
- Counts make the taxonomy auditable: a category with two transactions is a candidate for merging, and that is invisible without them.
- Changing a nature changes what the app calls spending versus investing (§8.3), so the consequence is stated where the change is made.

16 · Ledger — reference, not decisions

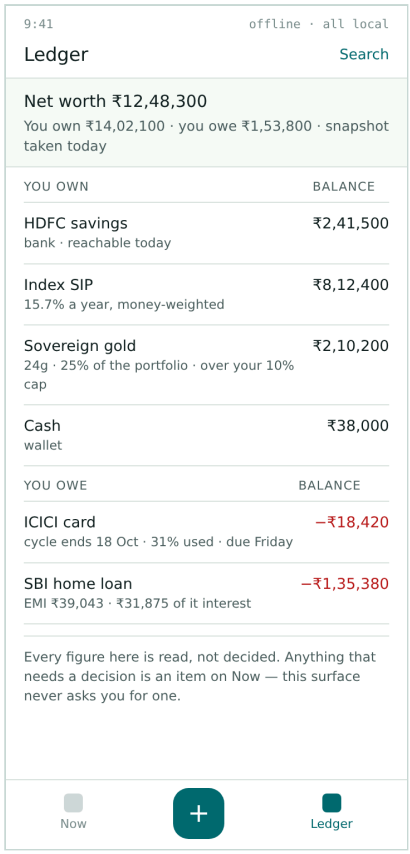
the second surface · principle 1



v1 — ships today



v2 — one answer per screen



v3 — the forward ledger

> **v3 vs v1:** the four buttons per row are gone and the rows group into own/owe
>
> **v3 vs v2:** tables rather than cards, and the surface states that it never asks for a decision

The idea

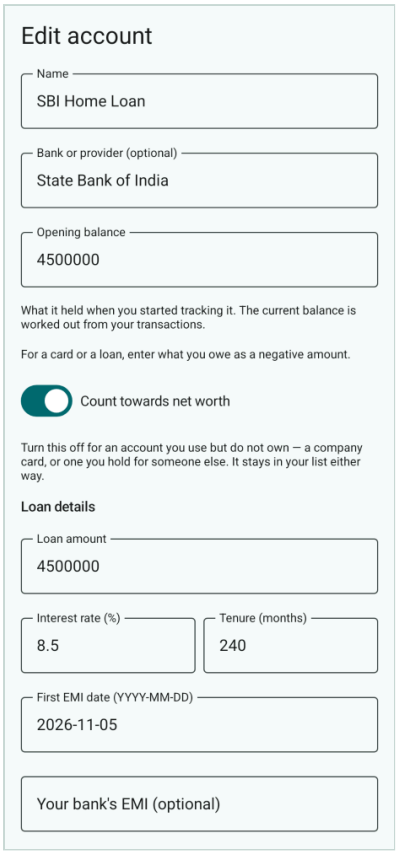
- Two tables: what you own, what you owe, with net worth as the counter above them.
- Sub-lines carry what the engines know — utilisation, how much of an EMI is interest, a concentration over its cap.
- A closing line states the surface's job: it never asks for a decision.

Why, and what it costs

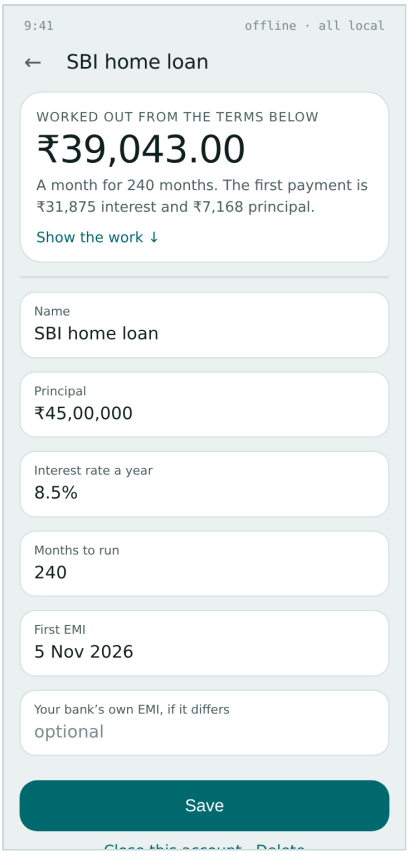
- Splitting reference from obligation is what keeps Now short. If balances lived on Now, the queue would be buried under things that are merely true.
- A table is denser than cards and is the form people already use for balances; at twenty accounts the difference is a screen against a scroll.
- Everything that *would* need a decision — the card due Friday, the gold over its cap — appears here as a sub-line and on Now as an item. The duplication is deliberate: one place to look things up, one place to act.

17 · Account editor

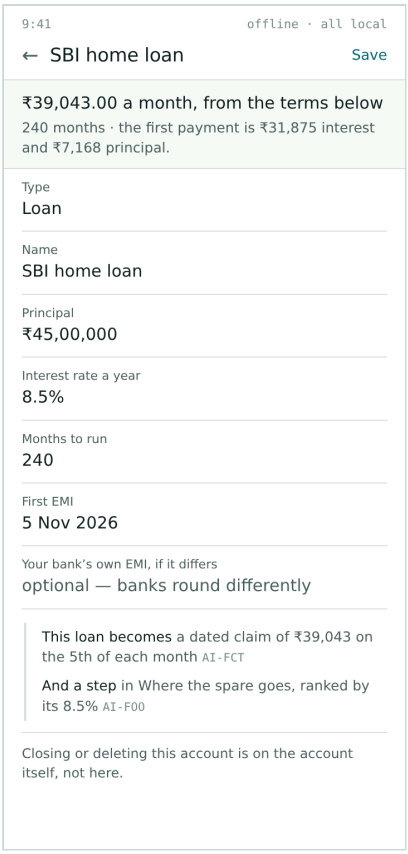
FR-ACC-002/003 · principle 2



v1 — ships today



v2 — one answer per screen



v3 — the forward ledger

> **v3 vs v1:** the EMI the terms produce leads the screen
>
> **v3 vs v2:** adds where this account shows up next — a dated claim, a step in the order

The idea

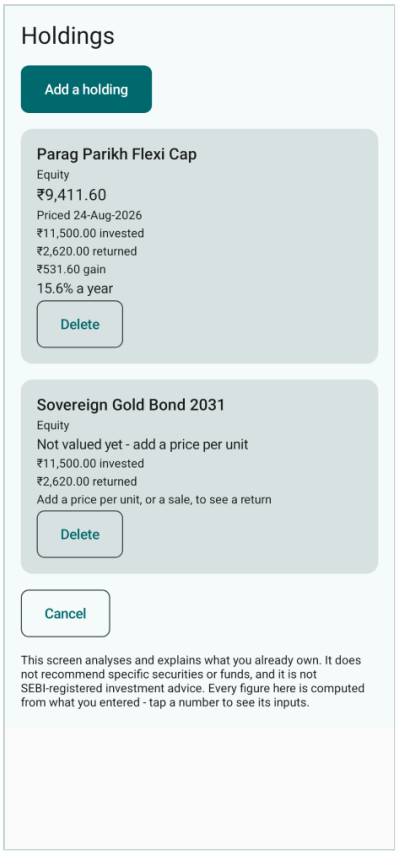
- The derived EMI is the counter, with the first payment's interest/principal split.
- A trail says what this account becomes elsewhere: a dated claim in the forecast, a step in the order.

Why, and what it costs

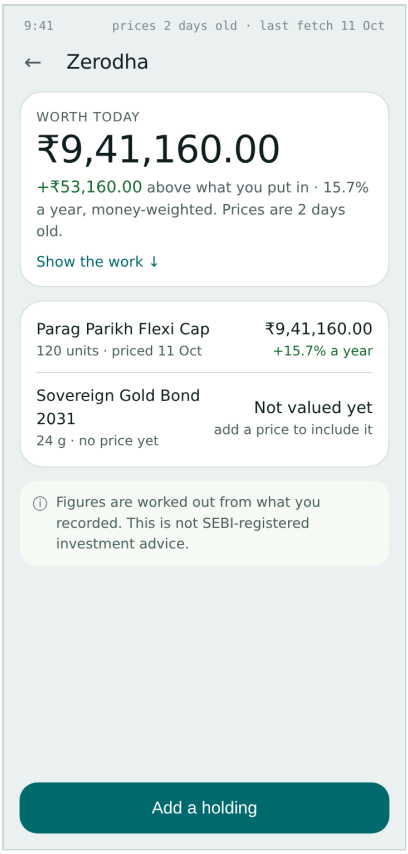
- v2 showed the EMI the terms produce; v3 adds where it goes next, because in a queue-shaped app the useful question about a new account is what it will put on your list.
- Closing and deleting live on the account, not in its editor — a form should not be able to destroy the thing it edits.

18 · Holdings

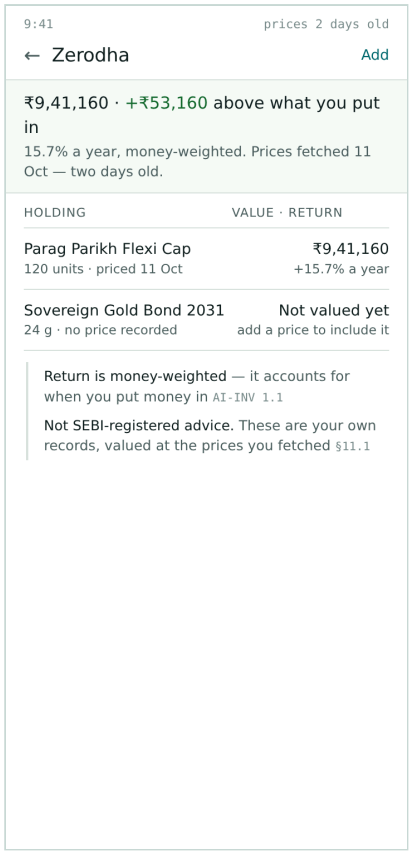
§11, AI-INV · principles 2, 4



v1 — ships today



v2 — one answer per screen



v3 — the forward ledger

> **v3 vs v1:** price age moves into the chrome and an unpriced holding says so
>
> **v3 vs v2:** the §11.1 disclaimer moves into the trail rather than sitting as a banner

The idea

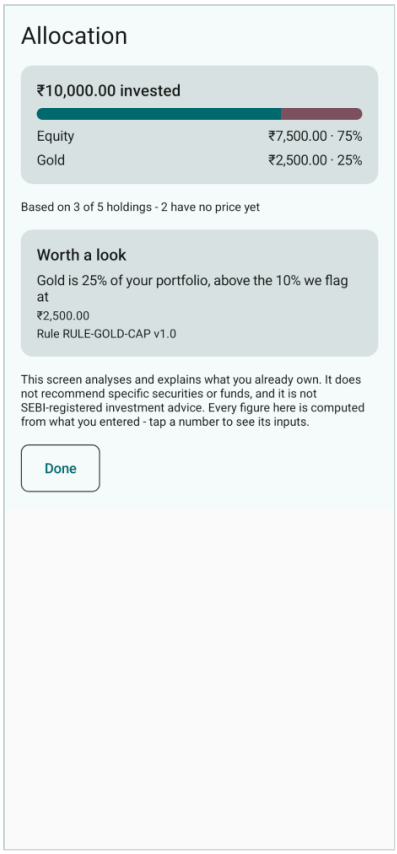
- Value and gain in the counter; price age stated there too.
- A table of holdings with value and return; an unpriced one says so and asks for a price.
- The SEBI position sits in the trail, with the rule that produced the return.

Why, and what it costs

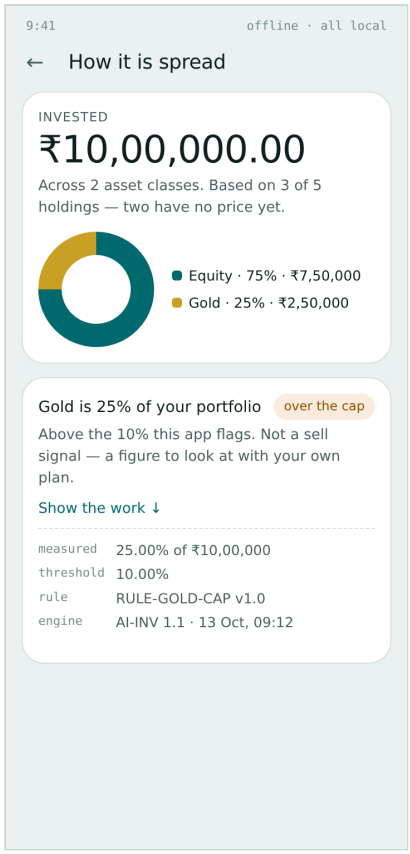
- A return and its basis belong in the same sentence (P-02); the trail is where v3 always puts basis, so the disclaimer reads as part of the working rather than as legal text bolted on.
- Staleness in the status line and the counter, because it qualifies every number below it.

19 · Allocation

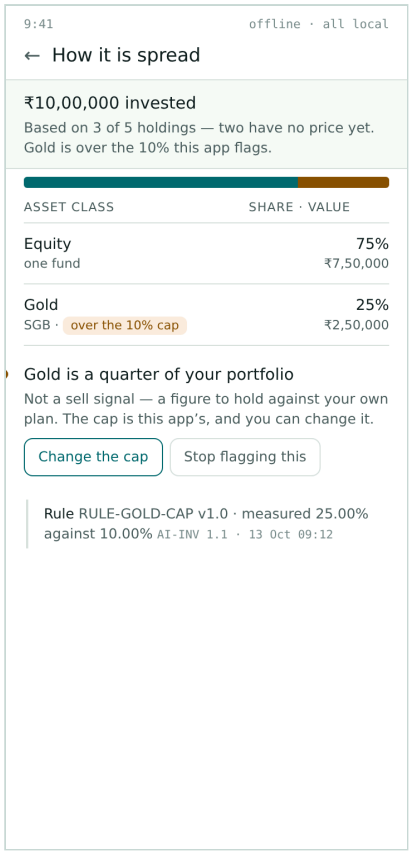
FR-INV-002, §11.2 · principles 2, 3



v1 — ships today



v2 — one answer per screen



v3 — the forward ledger

> **v3 vs v1:** the split is drawn and the flag carries its rule
>
> **v3 vs v2:** a bar and a table instead of a donut, and the cap becomes editable

The idea

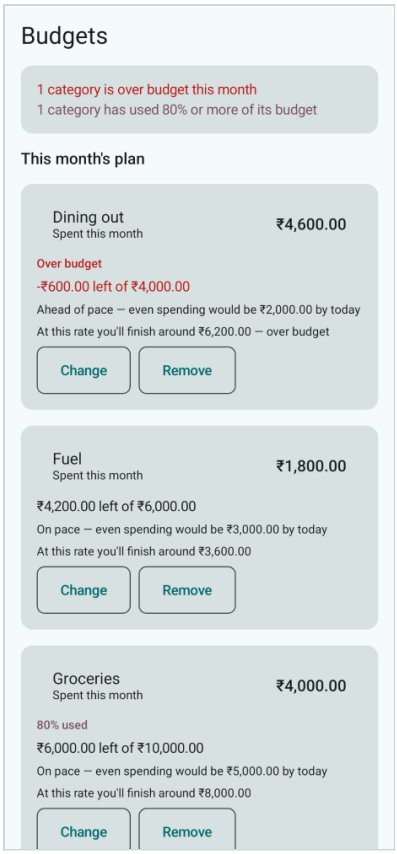
- A proportion bar and a table instead of v2's donut; share and value on every row.
- The cap breach is an item with *change the cap* and *stop flagging this*.

Why, and what it costs

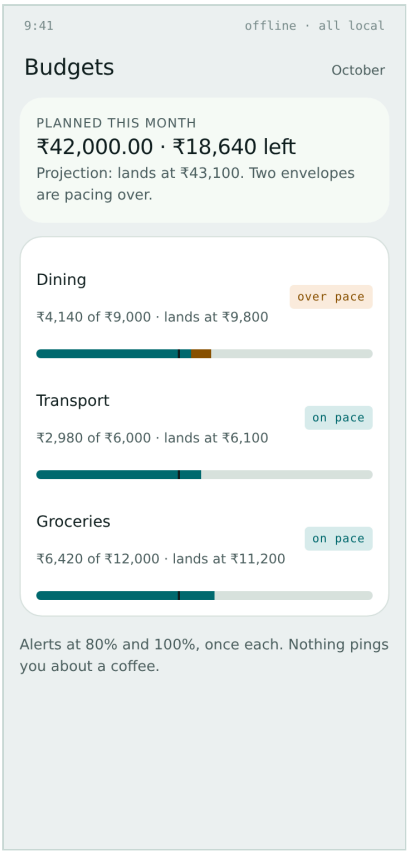
- v3's surfaces are paper, not canvas — a bar and a table carry a two-class split as well as a donut and survive 200% font better, which is where a fixed-size chart fails first.
- The cap is this app's opinion, so the user can move it or silence it. v2 says “not a sell signal”; v3 goes further and hands over the threshold (P-07).

20 · Budgets

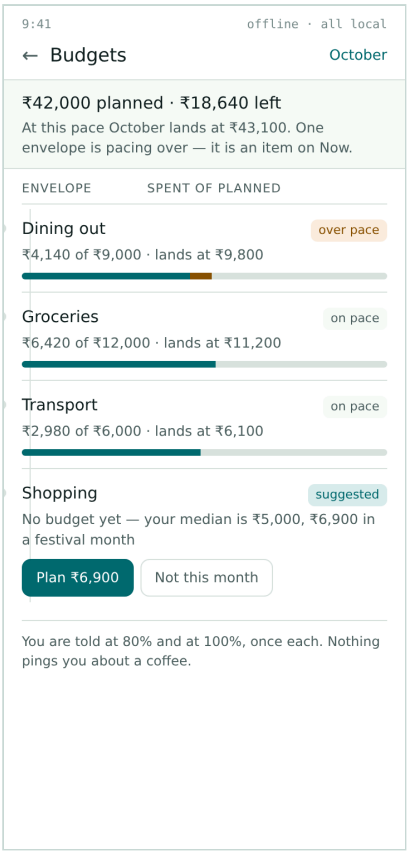
FR-BUD-001…004 · principles 1, 4



v1 — ships today



v2 — one answer per screen



v3 — the forward ledger

> **v3 vs v1:** pace is drawn once per envelope instead of described three times
>
> **v3 vs v2:** states its relationship to the queue: the alert lives on Now, this is where you plan

The idea

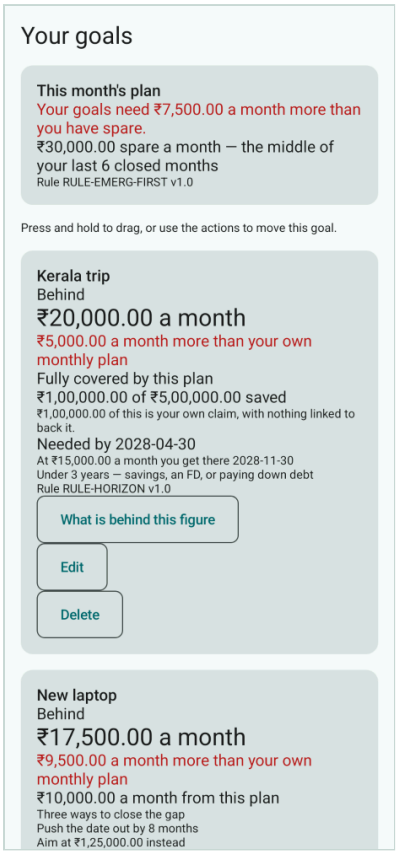
- The counter carries planned, left, and where the month lands.
- Envelopes are rows with the pace bar; the one pacing over says so and is already an item on Now.
- An unbudgeted category appears as a suggestion with its own two chips.

Why, and what it costs

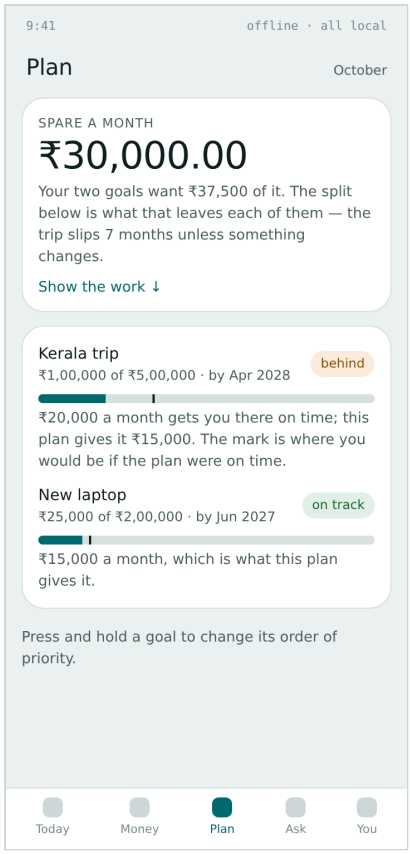
- v2's move 4 again, unchanged. What differs is that the screen states its relationship to the queue — the alert lives on Now, this is where you plan.
- Putting the suggestion in the list rather than on a separate card means the act of budgeting happens in one place, in one idiom.

21 · Goals as standing claims

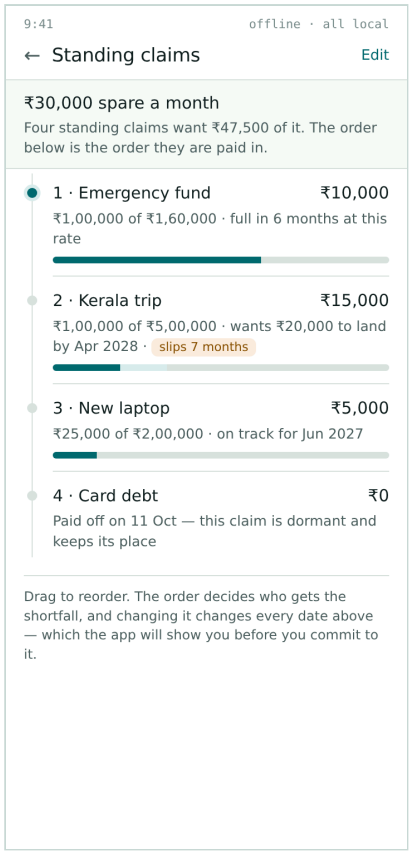
§15 and §36, restated · principle 1



v1 — ships today



v2 — one answer per screen



v3 — the forward ledger

> **v3 vs v1:** ten lines of prose per goal become a row, a bar and a pill
>
> **v3 vs v2:** goals are claims in priority order, not progress cards

The idea

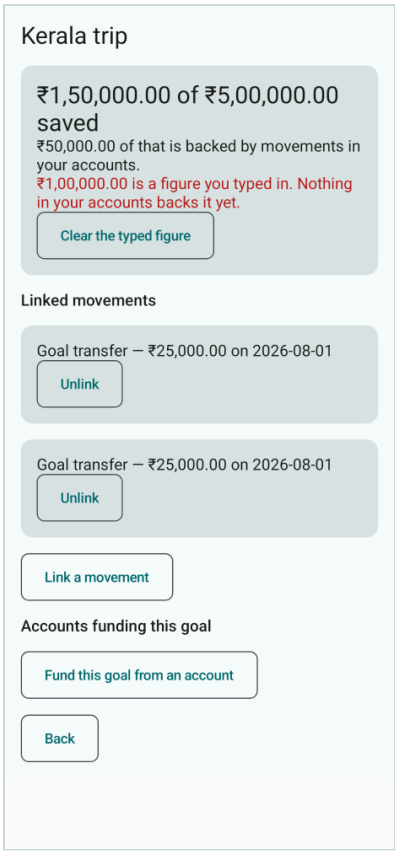
- Goals are a numbered list of claims on each month's spare, in the order they are paid.
- Each shows what it gets this month, what it wants, and what that does to its date.
- A dormant claim keeps its place in the order.

Why, and what it costs

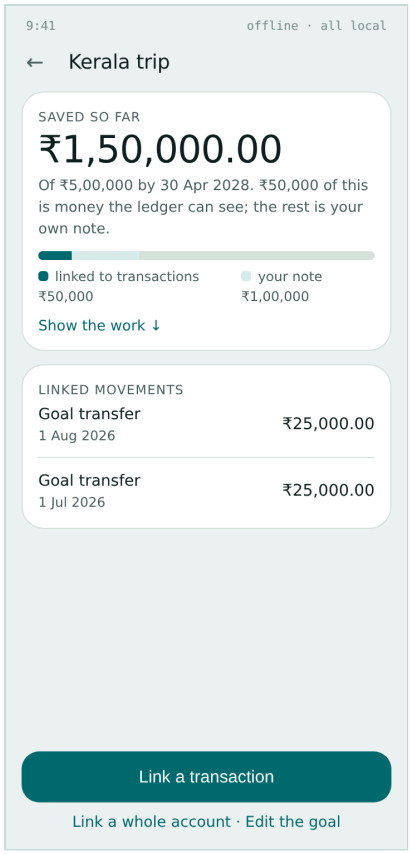
- This is what the waterfall engine actually does: it pours one month's surplus down an ordered list. v2 draws goals as progress; v3 draws them as claims, which is closer to the model and makes the ordering — the thing the user controls — the primary interaction.
- “Changing the order changes every date above, which the app will show you before you commit” turns a drag gesture into a decision with a preview (Nielsen #3, user control with a way back).
- The cost: a user who just wants to see how the trip fund is doing now reads a priority list rather than a progress bar. v2 is better for that user.

22 · One standing claim

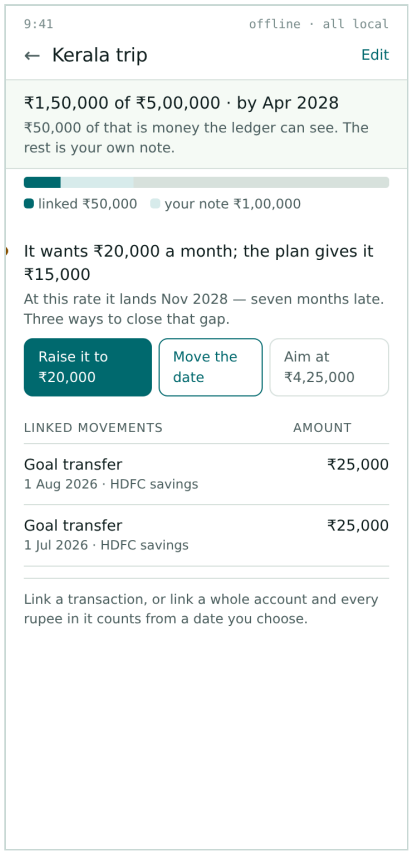
FR-GOAL-002/004 · principles 2, 3



v1 — ships today



v2 — one answer per screen



v3 — the forward ledger

> **v3 vs v1:** the evidenced/claimed split is drawn and named
>
> **v3 vs v2:** the gap becomes a decision with three ways to close it

The idea

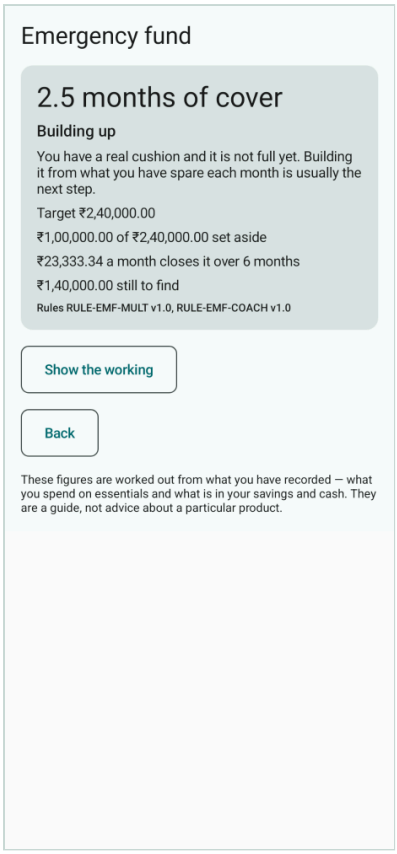
- The split between linked money and your own note is the counter and the bar.
- The shortfall is an item with the three ways to close it as chips.

Why, and what it costs

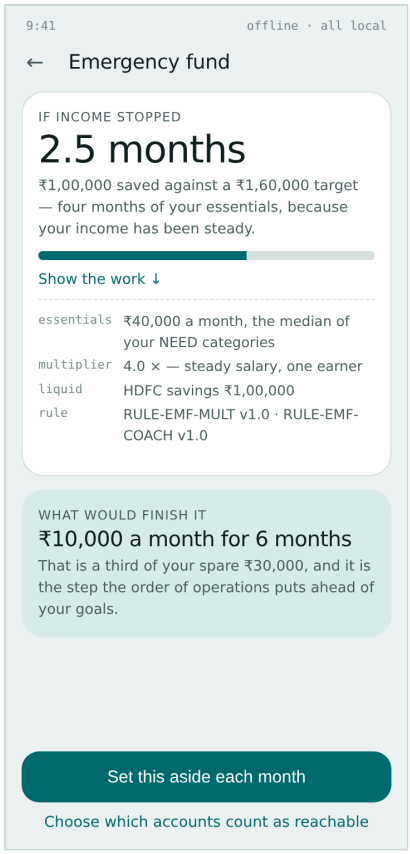
- Issue 7.4's distinction survives intact — it is the one thing this screen exists for — but the gap becomes a decision instead of a sentence, which is the v3 move.
- “Your note” rather than “unevidenced”: the user made the claim, and the app is not catching them out.

23 · Emergency fund

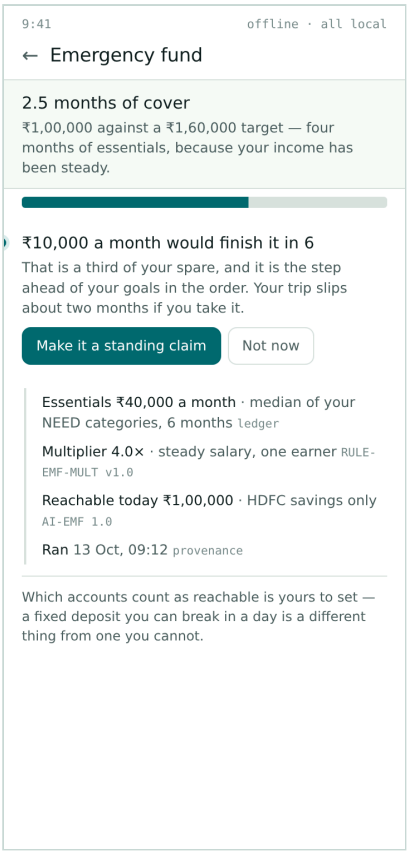
§10.1, AI-EMF · principles 2, 5



v1 — ships today



v2 — one answer per screen



v3 — the forward ledger

> **v3 vs v1:** months of cover lead instead of the target amount
>
> **v3 vs v2:** the coach line becomes an item that names what it displaces

The idea

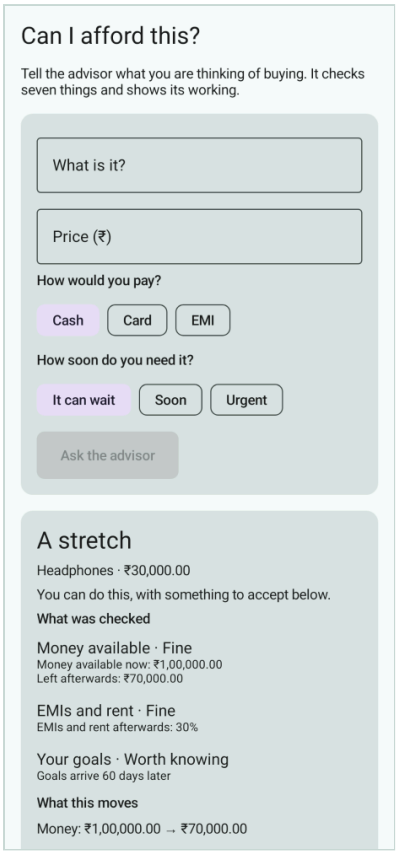
- Months of cover is the counter; the trail carries essentials, the multiplier and what counts as reachable.
- The coach line becomes an item: make ₹10,000 a month a standing claim, or not now.
- It names what that costs — the trip slips about two months.

Why, and what it costs

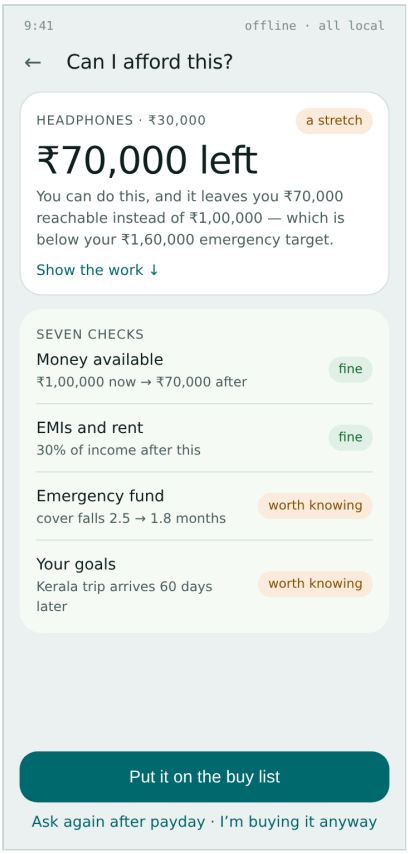
- §10.1 asks for evidence linked to every figure; the trail is where v3 always puts it, so this screen needs no special mechanism.
- Advice that ignores the other claims on the same rupee gets ignored. Saying what it costs is what makes the suggestion honest enough to accept.

24 · Can I afford this?

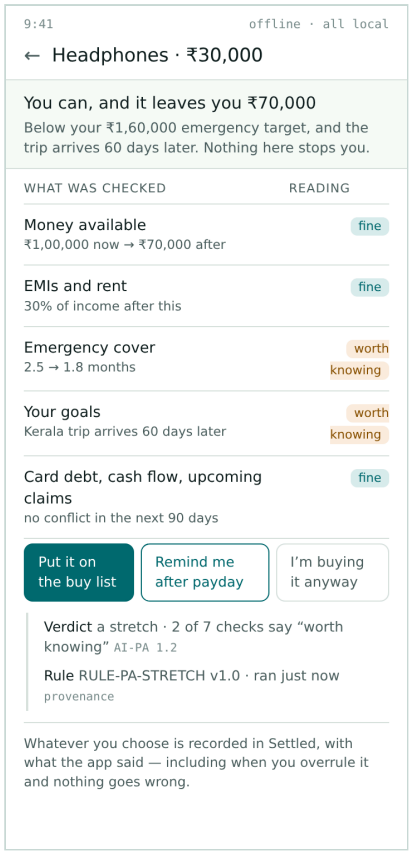
§13.2, FR-AI-003 · principles 2, 5



v1 — ships today



v2 — one answer per screen



v3 — the forward ledger

v3 vs v1: what the purchase leaves becomes the figure; the verdict is a chip

v3 vs v2: adds the record: whatever you choose goes into Settled

The idea

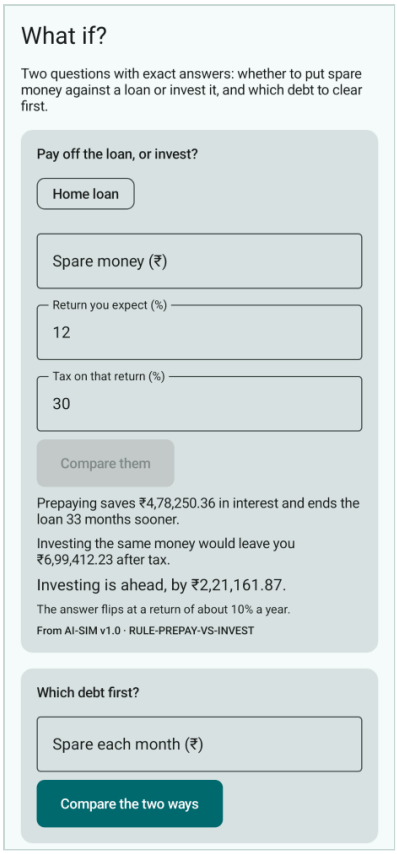
- The counter is what the purchase leaves you, not the verdict word.
- Seven checks as a table with readings and flags; three actions including *I'm buying it anyway*.
- A closing line says the choice goes into Settled either way.

Why, and what it costs

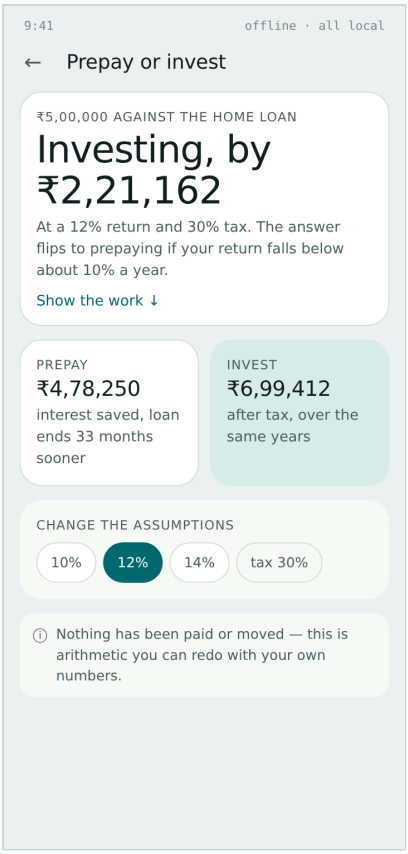
- P-07 is strongest here: the app's own actions are to remember or remind, never to approve. The user's override is a first-class option, not a way out.
- Recording the override is the v3 difference — if you overrule the app and nothing goes wrong, Settled says so, and the advice is judged on that.

25 · Prepay or invest

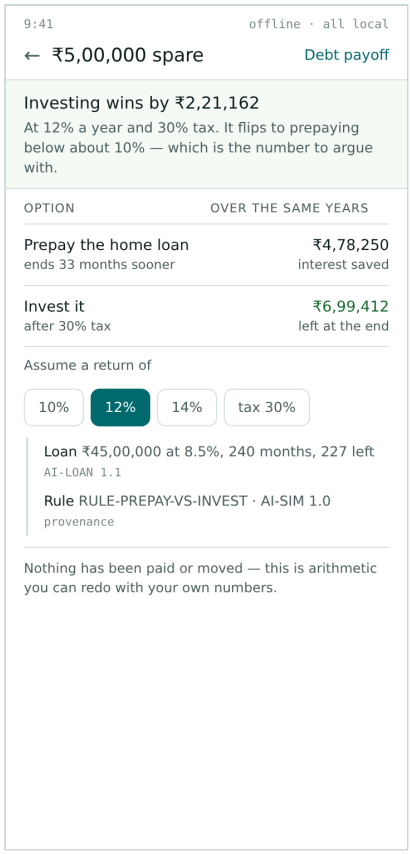
§36, §40.2, AI-SIM · principle 2



v1 — ships today



v2 — one answer per screen



v3 — the forward ledger

> **v3 vs v1:** both sides and the gap lead, with the flip rate stated
>
> **v3 vs v2:** a table rather than two cards, and the assumptions become chips

The idea

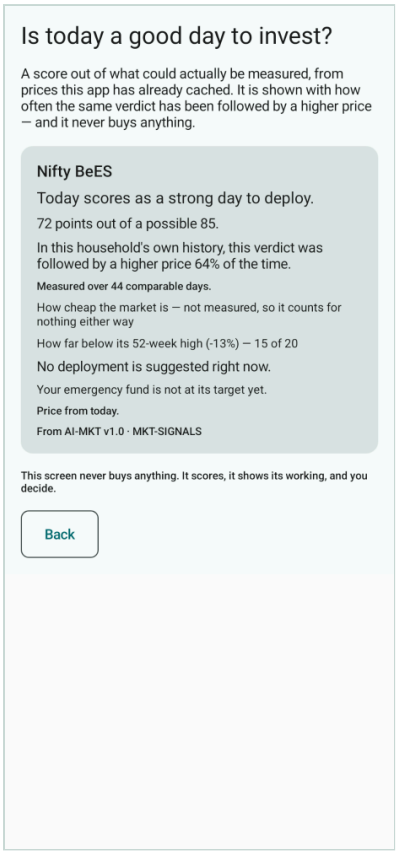
- The answer and its size in the counter, with the rate at which it flips.
- Both options in one table; the assumptions as chips you can change without a keyboard.

Why, and what it costs

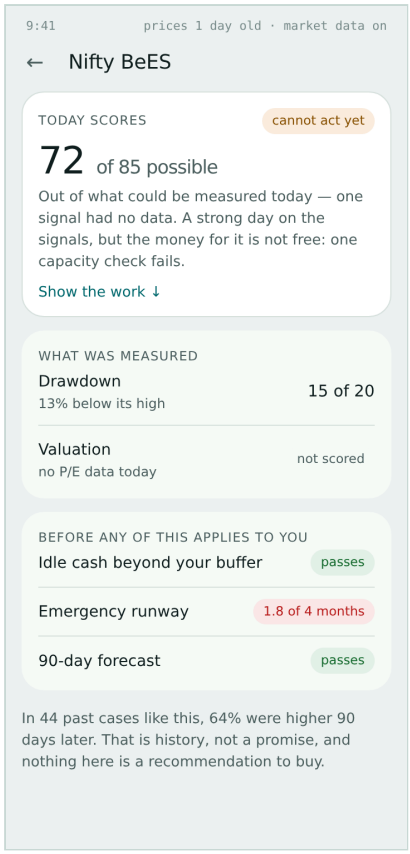
- Both sides visible or it is a verdict (P-02). A table makes the comparison literal — same rows, same units.
- The flip rate is the decision-useful number and the honest one: it tells you what you are really betting on.

26 · Opportunity

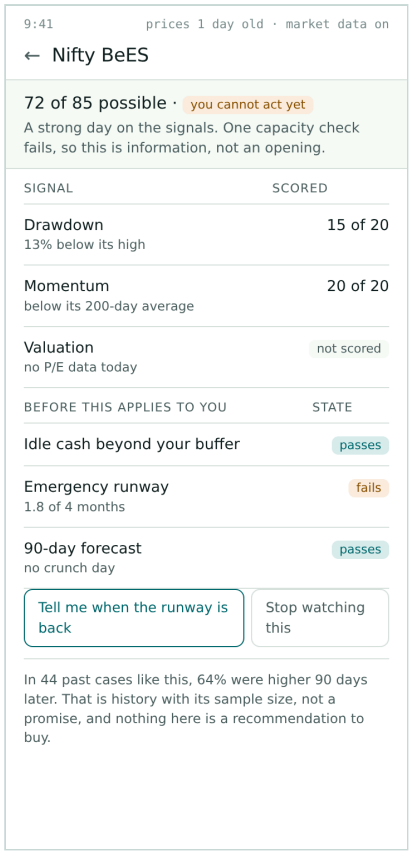
§30, AI-MKT · principles 2, 3



v1 — ships today



v2 — one answer per screen



v3 — the forward ledger

v3 vs v1: the score is out of what was possible and the gates are visible

v3 vs v2: adds a reminder action — leave with a condition rather than a feeling

The idea

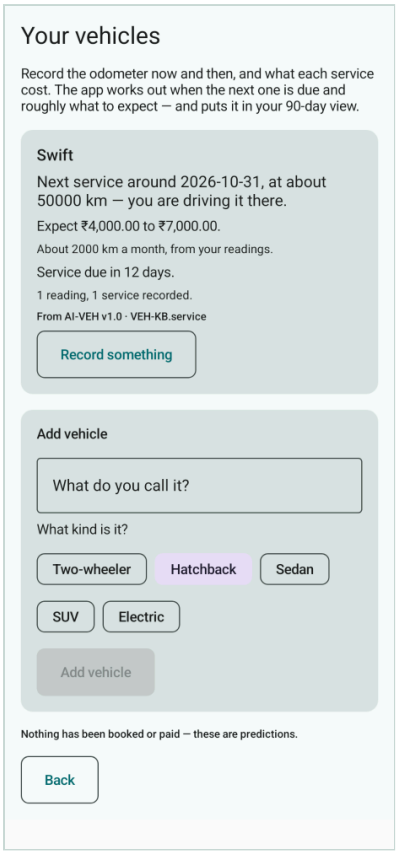
- Score out of what was possible, with the capacity verdict in the same line.
- Two tables: what was measured, and what has to be true before it applies to you.
- *Tell me when the runway is back* turns the screen into a future item.

Why, and what it costs

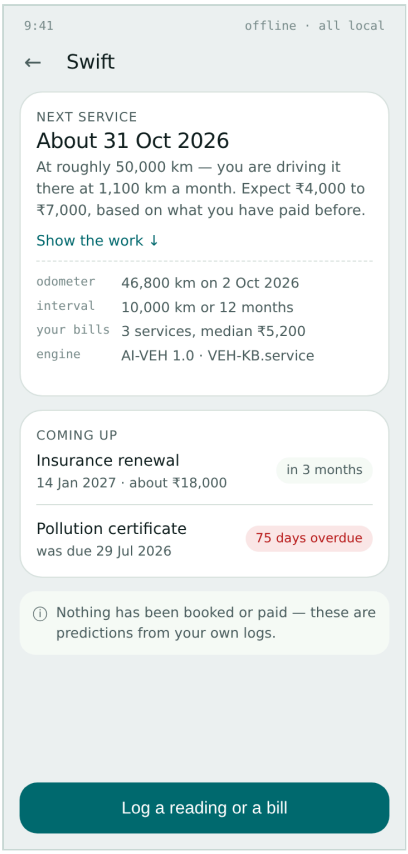
- The failing capacity gate outranks the score, because this is the screen most likely to be read as a tip.
- The reminder is the v3 answer to a market screen: nothing to do today, so leave with a condition rather than a feeling.

27 · Vehicle

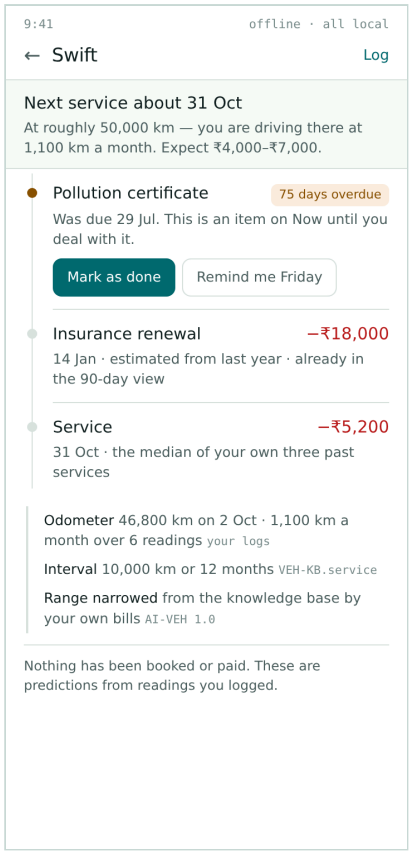
§12, AI-VEH · principles 1, 4



v1 — ships today



v2 — one answer per screen



v3 — the forward ledger

> **v3 vs v1:** the prediction carries its basis and overdue is a word with a number
>
> **v3 vs v2:** the overdue renewal is the same object as the item on Now

The idea

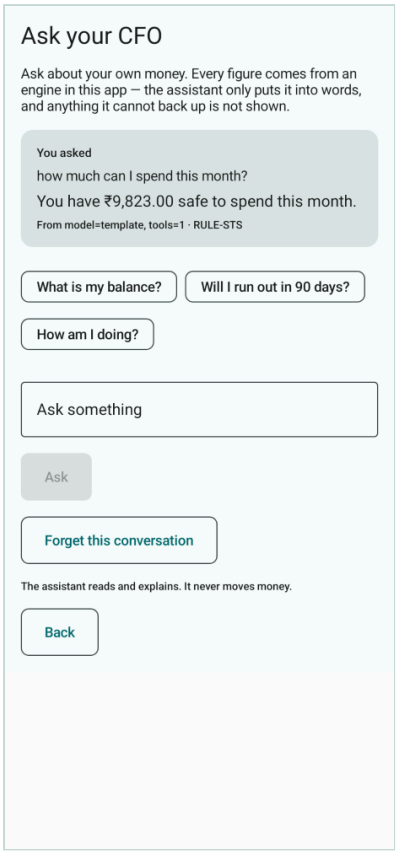
- Next service in the counter with its cost range; renewals as items, overdue first.
- The trail carries the odometer readings, the interval and whose bills narrowed the range.

Why, and what it costs

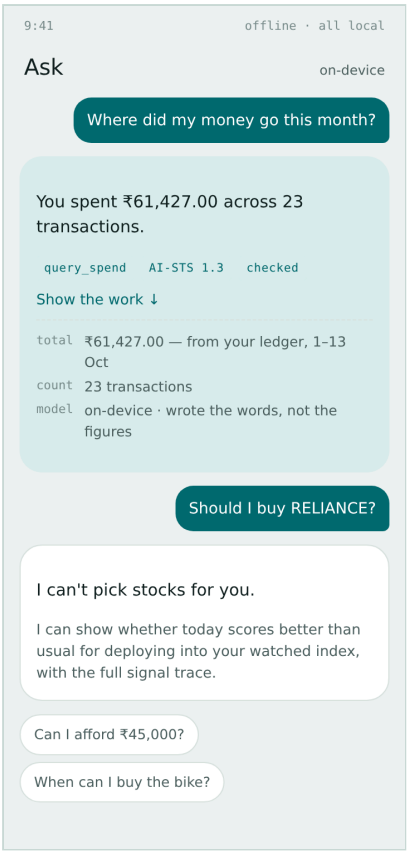
- The overdue certificate is already an item on Now; here it keeps the same shape, so the user meets one pattern twice rather than two patterns once.
- Naming the three inputs is what makes “about ₹5,200” arguable rather than magic.

28 · Ask

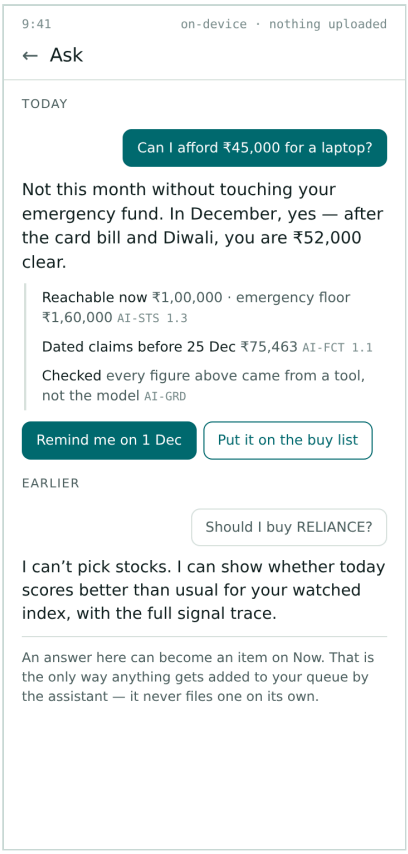
the assistant, inside the ideology · principles 1, 3



v1 — ships today



v2 — one answer per screen



v3 — the forward ledger

> **v3 vs v1:** a conversation layout with the composer pinned and the trace on the answer
>
> **v3 vs v2:** an answer can become an item, and only the user files it

The idea

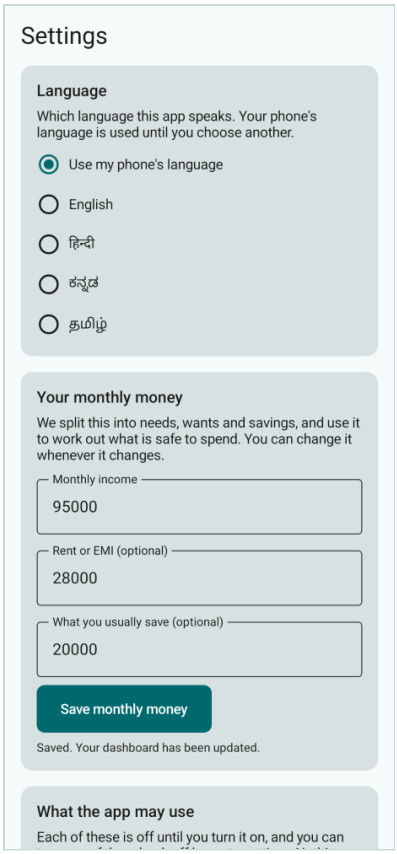
- An answer can become an item — *remind me on 1 Dec* — and that is the only way the assistant adds to the queue.
- The trail sits under the answer: which tools ran, which engine, and that the figures came from tools rather than the model.
- A refusal still offers the adjacent capability.

Why, and what it costs

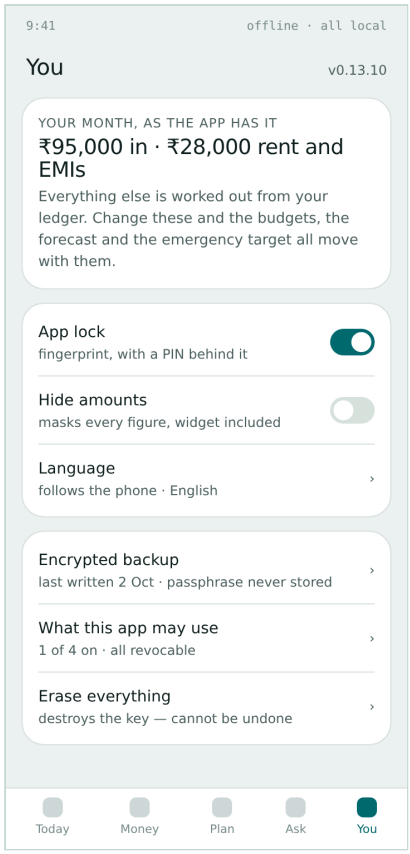
- If the queue is the app, the assistant's useful output is a queue item, not a paragraph. “Not this month; in December, yes” is only actionable if it can be made to come back in December.
- “It never files one on its own” is the restraint that keeps an agentic feature inside P-07. The user accepts each item explicitly.
- The guardrail line (AI-GRD) stays on the answer, because in a queue the answer is read once and then acted on — there may be no second visit in which to open a disclosure.

29 · You

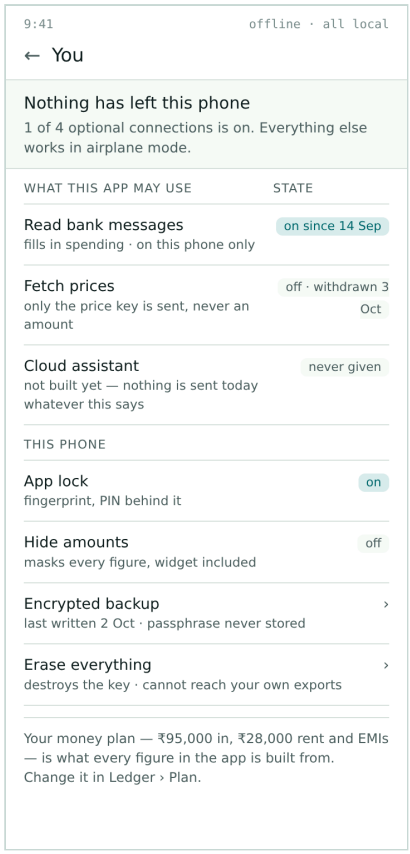
the standing promises · principle 5



v1 — ships today



v2 — one answer per screen



v3 — the forward ledger

> **v3 vs v1:** grouped by what each control changes, with the backup warning where it is acted on
>
> **v3 vs v2:** tables, with every state visible without opening a row

The idea

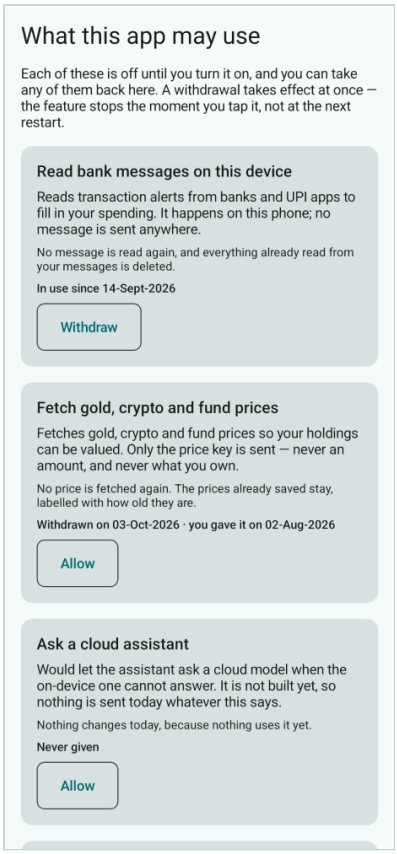
- The counter is the promise: nothing has left this phone, 1 of 4 connections on.
- Two tables — what the app may use, and what this phone does — with state visible without opening a row.
- Both dates stay on a withdrawn consent.

Why, and what it costs

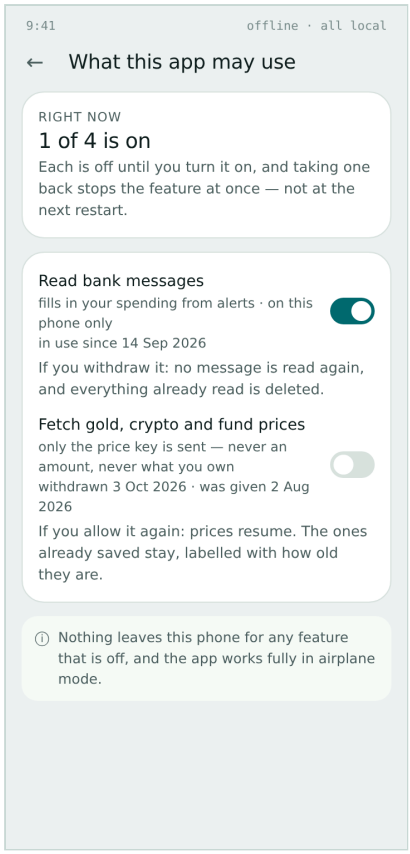
- Same content as v2, denser presentation, because this screen is read as an audit and audits are read in tables.
- Keeping state in the right-hand column means the privacy posture can be checked in one look, which is what “revocable, per feature” (P-01) deserves.
- The money plan moves to a line at the foot pointing into Ledger: in v3 it is reference data, not a setting you change often.

30 · What this app may use

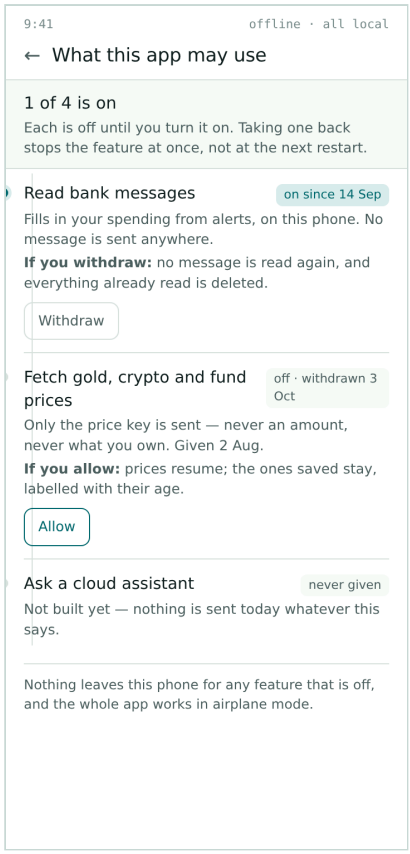
§23, P-01, DPDP · principles 3, 5



v1 — ships today



v2 — one answer per screen



v3 — the forward ledger

> **v3 vs v1:** each row gains what stops if it is withdrawn, beside the switch
>
> **v3 vs v2:** each consent becomes an item with its action inline

The idea

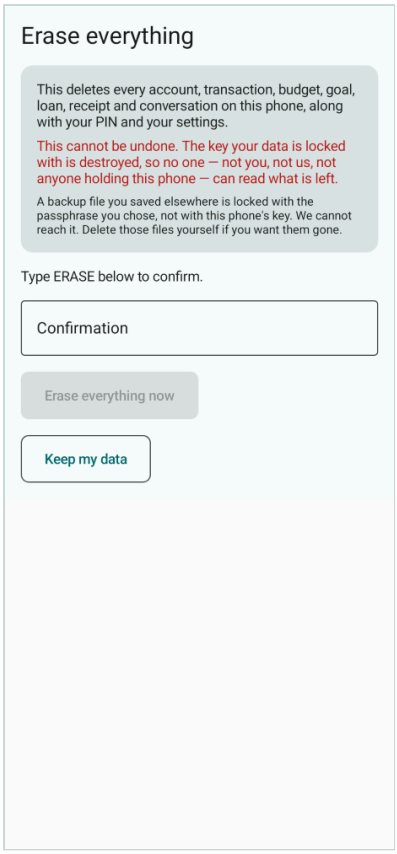
- Each consent is an item: what it does, since when, what stops if withdrawn, and one action.
- A withdrawn consent keeps both dates and the way back.

Why, and what it costs

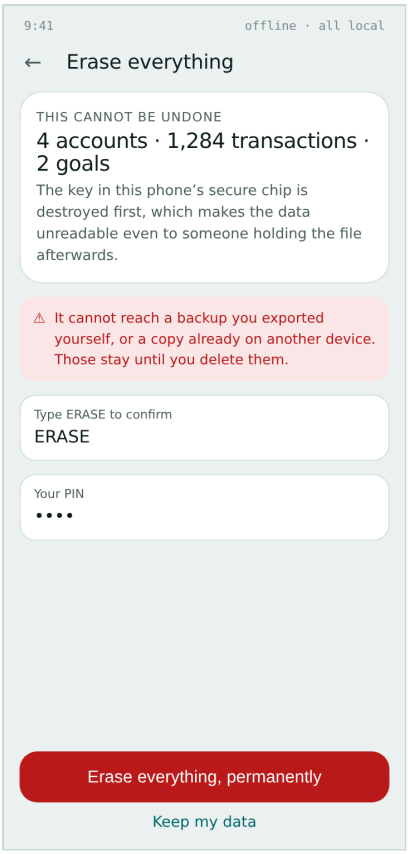
- A consent is a standing decision, so it gets the item treatment — including the fact that you can change it here and now.
- Both dates stay because this is the evidence the promise was kept (P-01, DPDP), and v3 is the design that keeps records of decisions.

31 · Erase everything

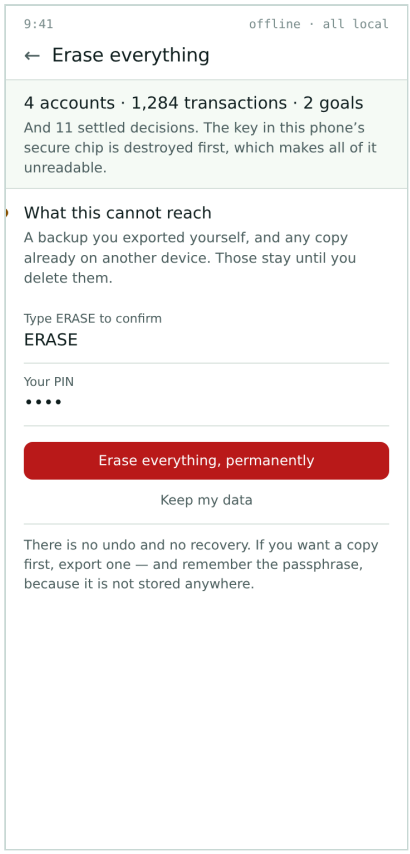
§23, §34, SEC-003 · principles 2, 4



v1 — ships today



v2 — one answer per screen



v3 — the forward ledger

> **v3 vs v1:** the scope is counted and the caveat is the loudest thing on the screen
>
> **v3 vs v2:** the count includes the settled decisions v3 itself created

The idea

- The counter states the scope, including the settled decisions that go with it.
- What the erase cannot reach is an item above the fields, not a footnote.
- One destructive action, in the negative role, with the alternative as plain text beneath.

Why, and what it costs

- “Everything” is not a scope; four counts are. The decision log is in the count because v3 created it, and the user should know it dies here too.
- The one thing that cannot be discovered afterwards is what survived, so it is the loudest thing before.

What v3 risks

Every one of these is a real cost, and none of them is fixed by better drawing.

Risk	Why it is real	What would have to be true
A queue implies obligation	Money is not a to-do list. A permanent list of things you have not done is a different emotional object from a dashboard, and for an anxious user it is worse.	The empty state has to be common, and the app must be comfortable saying “nothing needs you” most days.
Learned dismissal	If items appear that do not deserve a decision, users learn to clear the queue without reading it — and then miss the one that mattered.	A high bar for what becomes an item, and the no-repeat rule enforced in the engine, not the UI.
Settled needs new storage	There is no decision log in the schema today. Recording what the app said and what the user did is a new table, a migration, and a retention question.	An ADR, a schema version, and a migration test — and a decision about whether that record is exportable and erasable (it must be).
Discoverability	A feature that never produces an item can become invisible. The skill already warns that a screen only the developer can find is a screen nobody uses.	Ledger must hold every screen, and the three-taps-from-a-surface rule still applies. The queue is a fast path, never the only one.
Density at 200% font	v3's rows carry more text than v2's cards. Nothing has a fixed height, but a two-line item becomes six lines at 200%, and three buckets of those is a long scroll.	Baselines at 200% in all four languages before anyone commits to this. It is the test v3 is most likely to fail.
It loses v2's calm	Checking net worth now costs a tap into Ledger, and the home screen is busier by design.	Accepting that the app is for acting, not browsing — which is a product decision, not a design one.

When v3 is the wrong answer

- **When the user already has their money under control.** A queue that is empty six days a week argues for itself; one that the user has to be taught to care about does not.
- **When the team cannot afford the decision log.** Without *Settled*, v3 is a to-do list with good manners — principle 5 is what makes it more than that.
- **When notifications are already a sore point.** The queue makes it tempting to interrupt; §17.2's restraint has to hold or the ideology turns into nagging.

How to choose, without arguing about taste

The two designs make different predictions, and both are testable on the same build in a week:

- **v2 predicts** that people open the app to check, so the win shows up as faster comprehension: time to answer “can I spend this?” and “how much is left in dining?”
- **v3 predicts** that people open the app to act, so the win shows up as decisions: items acted on per week, overdue items reaching their deadline, and — the honest one — **sessions that end with nothing done, which v3 should make *more* common, not less**.
- **Both** should be measured against the same accessibility bar: 200% font, dark, and the privacy blur, in English, Hindi, Kannada and Tamil.

> **They are not exclusive.** The honest hybrid is v2's five tabs with *Now* as the first one, keeping Today as the second — which gets the queue without giving up the calm home screen. That hybrid is worth naming because it is probably where this ends up, and because choosing it deliberately is better than drifting into it.

The mockups are editable HTML in `docs/design/redesign-v3/mockups/`, and the stylesheet holds v2's tokens verbatim — so any change that looks wrong there is wrong in the app too. Nothing in this document changes a colour, a token, a string or a line of application code.