



HI-5 Global Challenge Coder's Playbook

Building the world's first Humane Intelligence, by the people, for the people.

Version: 2.1

Date: 12 January 2026



Contents

1	The Starter’s Gun Has Already Fired	5
2	Build Primer — What You’re About to Build.....	6
2.1	The Activation Constraint	6
2.2	What You Are About to Build	7
2.3	The Architecture’s “Physics”.....	9
3	Team Formation: Your First Fork in the Road.....	10
3.1	Choosing the People Who Can Survive the Journey	10
3.2	Early Partnerships: Why Storytellers Cannot Be an Afterthought.....	11
3.3	Strategic Positioning for the Bonus Categories.....	12
3.4	Reading the Collaboration Hub and Choosing Allies Wisely	13
3.5	Your Story Starts Now, Whether You’re Ready or Not.....	13
4	Project Setup: Spin Up Fast or Fall Behind.....	14
4.1	Your Technical Battlefield	14
4.2	Environment Setup: Erasing Chaos Before It Starts	15
4.3	Planning Without Overplanning.....	16
4.4	The Early Gotchas That Will Gut Your Momentum.....	16
5	The Building Phase: The Decisions That Decide Everything	17
5.1	Turning the Specification Into AI Power	17
5.2	The Protocol Dilemma: Build It Yourself or Partner?	18
5.3	Collaborating Without Losing Your Advantage	18
5.4	Storytellers: What to Reveal, What to Hide, and When to Use Them as a Weapon	19
5.5	Tracking Contribution Before It Becomes a Fight.....	20
5.6	The Team Lead’s Burden	21

5.7	Simulation: Where Vision, Data, and Drama Collide	22
5.8	Protocol: The Category With No Mercy	24
5.9	Mobile App: The Category Everyone Underestimates	25
5.10	Desktop App: Power, Performance, and Zero Room for Weakness.....	27
6	The Submission and Judging Phase	29
6.1	The Merger Challenge: What Happens Before the Event	29
6.2	The Finalist Showcase: What Happens During the Event.....	30
6.3	The Second Chance Challenge	32
6.4	The Bonus Category Event: The Final Public Showcase	34
6.5	Life After the Challenge.....	35
6.6	Positioning Early for the Bonus Categories.....	36
	Appendix A: The Secret They Never Wanted You to Know	38
	Love — the Architecture of Real Intelligence	38
	Trust — the Nervous System of the Swarm.....	38
	Forgiveness — the Secret to Eternal Growth	39
	Why They Hid It	39
	The Age of Humane Intelligence™	39
	Join the Revolution of Heart and Code	40
	Appendix B: HI-5 Grand Challenge Schedule	41

1 The Starter's Gun Has Already Fired

If you're reading this, you're late! The starter's gun has fired and is now cold. Other teams didn't wait — they just dropped straight into sprint mode. Late developers build differently. They build fast. Hard. With intent.

This is the underground world where devs become contenders, where wit and grit matter more than perfect planning, and where anyone can take the lead if they're willing to burn a little hotter.

The challenge is already unfolding. Teams are already merging, splitting, scheming, collaborating, and preparing for the shortlists and finalist showcases.

Your rivals are running hard. Your job now is simple - catch them, pass them, own the category they thought they had.

This Playbook is your compass, your lifeline. If you've got what it takes, then it'll help you mobilize your potential to out-deliver the teams who think they've already claimed their spot on the podium.

Settle in, spin up your environment, stretch your arms and grab that oversized mug of coffee because you're going to be here for a while. Time to make up ground. Time to enter the race for the million dollar prize, where basement legends rule.

2 Build Primer — What You're About to Build

mypaloma isn't a normal app. It's a swarm-based distributed system where many small AI models ("Sprites") interact, and the user experience depends on the collective behavior of the swarm rather than any single model's performance.

The HI-5 Challenge is built to generate two outcomes in parallel: the technical components required to launch, and the mobilization required to reach "activation" viability.

In the subsections that follow, you'll get a high-level view of what you're building and where your category fits; the core concepts that underpin its architecture (swarm behavior, governance constraints, and incentives); and a translation layer that takes loaded terms like *love*, *trust*, and *wellbeing* and expresses them as formalized, testable definitions rather than vibes. The mypaloma spec is explicit that these concepts are treated as "measurable and actionable criteria" and implemented as agent DNA attributes.

Before we talk architecture, you need to understand the primary constraint, because it sits above everything else.

2.1 The Activation Constraint

Most products can launch as a controlled pilot, iterate quietly, then scale. mypaloma cannot. Think of it like a beehive – individual bees can sting you, yes, but when they work together as a swarm, their attack can be devastating. This 'swarming' behavior is the heartbeat of mypaloma, and its true potential. Without it, mypaloma is simply lots of small AI models that have no real capacity to provide value to their user.

For mypaloma to swarm, two things need to happen: one, there needs to be a large enough number of Sprites; and two, they need to 'turn on' at the same time to activate the swarm.

In that framing, launch isn't a release date. Launch is a birth event. It's the first moment the system behaves like a swarm, which requires a coordinated activation event where enough Sprites begin interacting from a single point in time and

continue thereafter. If that activation doesn't cross the threshold, you don't just get a "weak launch", you get a "failed birth". Either the system crosses the threshold and exhibits swarm behavior, or it doesn't.

The uncomfortable part is that the minimum viable swarm size is unknown. You can't theoretically calculate this threshold, nor can you safely "discover" it using a "start small" approach.

This has direct implications for builders, even if you think you're "just coding." HI-5 isn't "marketing as persuasion"; it's a mobilization engine under a hard systems constraint. The project needs enough committed early participants to support coordinated activation at launch, and it needs to hold attention long enough to get there. That means your component should not only be demo-able in isolation, but also legible, reliable, and integration-friendly under simultaneous participation.

One terminology note will save a lot of unproductive debate. You'll hear "cold start," which is nothing more than the first members, the "Genesis Members", being the "parents" to birth and raise this "2-year-old" AI swarm. It is their job to be patient and teach it human values. These differing values will start to create mini swarms that mimic societal cultures and subcultures, representing the start of digitally mirroring humanity's complex diversity.

You will also hear terms such as "love", "wellbeing", "trust", "forgiveness" and so on. Make no mistake, it took 2 years of research to convert these philosophical terms into measurable and codable constructs. All philosophical constructs are defined in incredible detail in the specification so that they are easy to code and test. Yes, mypaloma may be a world first in so many ways, and maybe one day, you can look back and know that your code is somewhere "floating" around in the swarm, making it all work.

2.2 What You Are About to Build

At the center is the idea that Sprites live on edge devices—phones, tablets, laptops, desktops. In practical terms, the mobile and desktop apps are not "clients" in the

usual sense; they are homes for Sprites and the user's portal into the ecosystem. This is critically important, so it is worth pulling out and repeating:

“The mobile and desktop apps are homes for Sprites and the user's portal into the swarm.”

Because there's no central coordinator, Sprites rely on robust peer-to-peer communication to exchange information, learn, and replicate across devices, which is why the protocol work is described in the following sections as the “lifeblood” of the system.

The **Simulation** serves 2 critical purposes. Firstly, it is the engine that uses a special type of “immortal” Sprite called a “Primordial” to organize and associate data to a massively diverse set of human behaviors. Once this is done, the Primordial Sprites then produce the first set of real Sprites called a “Genesis” Sprite. These Genesis Sprites are released to special set of “Genesis” users that have accepted the responsibility of activating the swarm and teaching the newly born swarm human values. And secondly, the simulation tests how these Genesis Sprites swarm by simulating their interactions over just a few days. This allows the data scientist teams back at the PALOMA headquarters to rerun the simulation over many extreme scenarios to help ensure that the swarm will stay ethically aligned with humanity.

The **Protocol** is the P2P communication mechanism that makes the swarm “fully decentralized and distributed with zero single points of failure”, alongside latency/performance and security/robustness.

The **Mobile App** is the main user experience when interacting with the swarm. It is the primary gateway into the swarm.

The **Desktop App** is where greater compute is leveraged to support more powerful Sprites and advanced functionality.

Finally, the “Storyteller” teams are not a gimmick – if you don't have one, you will probably not win. Your first step is either to find one of these teams in your local area and convince them to team up with you. And if there are no Storyteller teams near

you, then find 3 people to create a new Storyteller team that can broadcast your story using their mobile phones. Once you have a dedicated Storyteller team, you must allow them to film and broadcast you and your team's drama over the next 10 months as a reality-TV-style series. During the semi-finals, the judges and live audience assess the Storyteller's broadcast, allocating a whopping 20 points (out of 100) of your score to how good their story is.

2.3 The Architecture's "Physics"

The system is designed to exclusively run on edge devices. Without a central server coordinating activity, Sprites must rely on peer-to-peer communication to exchange information, learn from each other, and replicate across devices. The intent is that learning can propagate through the swarm as Sprites share what they've learned with others, creating a "ripple effect" across the network that fuels emergent intelligence. To avoid communication broadcast storms, the design uses behavior to moderate how 'wide' and 'deep' a broadcast request needs to be. For example, a 'shy' or 'anti-social' Sprite will broadcast a question to a few other trusted Sprites, whereas a very 'social' Sprite will broadcast widely. A 'thoughtful' Sprite will set a long expiry time on a broadcast, whereas a 'non-thoughtful' Sprite will set it short. When these and other behaviors are used together, along with other secondary moderators, broadcast storms are architecturally avoided.

It is also important to note that a distributed system can still have central control points. A fully decentralized system like mypaloma has no central control, meaning that things like the control of broadcast storms, the control of research data collection, or the control of privacy vs sharing are controlled by the 'swarm' itself, not the individual Sprites. However, in saying this, it is the simple rules within each Sprite that allow this 'virtual' control to materialize within the swarm. Even simple things like how to defend against malicious Sprites being injected into the swarm? Again, this is controlled by the swarm's immune system that emerges from each Sprite's ability to 'trust' and 'forgive'. The swarm's ability to remember important information and forget non-truthful information emerges from a Sprite's 'wellbeing' and its ability to maintain its wellbeing by generating 'karma'. Again, the design specification details a Sprite's

behavioral DNA, and how it changes depending on the Sprite's environment (i.e. epigenetics), and how it passes its successful DNA and epigenetics to its offspring to improve the swarm through natural selection.

All of this leads to the point you raised about terms like *love*, *trust*, and *wellbeing*. In the specification, these are not treated as poetic labels; the document explicitly says these concepts are “measurable and actionable criteria,” implemented directly into each AI agent as DNA attributes that create diversity in interaction behavior and allow selection pressures to operate. If you wish to alter the design of a Sprite's DNA, then do so with caution. There is a reason it took 2 years research – even though it may seem simple on the surface, it goes deep into Complex Adaptive Systems Theory and Emergent Behavior Theory where a system's complexity arises through interaction simplicity.

3 Team Formation: Your First Fork in the Road

3.1 Choosing the People Who Can Survive the Journey

The first real decision you make in this challenge isn't architectural — it's human. And if you get this wrong, no amount of clean code or clever engineering will save you when the pressure hits.

Every team you're up against has already been forming since early February and March 2026, when registrations opened across the categories—Simulation on 4 February 2026, Mobile and Desktop on 18 February 2026, and Protocol on 4 March 2026. These weren't ceremonial dates. They were the moment clever teams began stacking their talent, shaping their chemistry, and bracing for the countdown to their respective submission deadlines: Simulation racing toward 6 May 2026, Mobile toward 3 June 2026, and Desktop toward 8 July 2026. They're already moving. You need to catch up.

The trap most late teams fall into is size blindness. You'll feel a pull in both directions — keep your team small and protect your eventual profit share, or grow fast to match the speed and skill depth of teams who've been building for months. The prize money isn't hypothetical. Winning teams receive real monthly profit-share distributions, split equally across all winning teams and founders. The team leader is responsible for dividing that inside the team. More people means more power. More power means more speed. But more people also means slicing your own prize thinner each month.

That tension is not your enemy. It is your calibration tool. The sweet spot is the smallest number of people who can still win your category — and the biggest number you can handle without losing cohesion.

Why cohesion? Because buried inside the rules is one of the most catastrophic triggers in the entire challenge: any team with a significant internal dispute at the time of shortlisting is automatically disqualified. No discussion. No appeal. One rogue personality or unresolved argument during the five-day shortlisting window after submission — and you're erased. Permanently.

Choose fast, but choose right. And once chosen, align. Your survival depends on it.

3.2 Early Partnerships: Why Storytellers Cannot Be an Afterthought

If you're a developer, your instinct might be to focus only on code. That instinct will cost you points, audience support, and possibly your entire category win.

Story is not optional. Not in this challenge. Every technical category requires at least a mandatory 5-minute video demonstration, and that video will be watched in two critical moments: once during shortlisting, once during the finalist showcase, and potentially by the global public when it is featured in the Kickstarter campaigns.

Bring storytellers in early, and your journey becomes a narrative weapon. Bring them in late, and you hand your competitors a free advantage.

Those teams who began the moment registration opened have already been documenting their progress. Think about what that means when the audience voting

window opens for exactly five minutes during the finalist judging event — the moment when the public contributes up to 30% of your total score. If your story hasn't been unfolding from Day One, the audience won't know who you are, won't know your struggle, won't know your breakthroughs — and won't vote for you.

This isn't vanity. It's strategy.

And storytellers don't only affect audience votes. Their presence positions you for the Best Broadcast Human Story bonus prize, and their embedded access to your team naturally increases your eligibility for Best Cross-Team Integration later. That early partnership creates pathways. Late partnerships close them.

So treat storytellers the way you treat architects and backend engineers — as critical members of your velocity, not decorative side characters.

3.3 Strategic Positioning for the Bonus Categories

You may think the bonus prizes — five separate \$200,000 awards — are something to think about later. That mistake will cost you dearly.

Every bonus category begins the moment your team forms. Not later. Not halfway. Not “once the main build is stable.” Now.

The Most Diverse Team category is shaped on the day you choose your people. The Best Open-Source Movement category begins with the structure and clarity of your repo on day one. The Best Innovation & Creativity award is built from the decisions you make at your earliest design discussions. And Best Cross-Team Integration is determined by whether you choose to isolate yourselves or collaborate intelligently from the beginning.

These aren't side prizes — they are parallel routes to victory. Teams who lose their main category can still walk away with large profit-share awards if they position correctly.

Your formation choices ripple outward for months. Treat them like the first lines of code in your system — foundational, irreversible, defining.

3.4 Reading the Collaboration Hub and Choosing Allies Wisely

When you step into the official Collaboration Hub, you'll feel the electric buzz of teams forming alliances, searching for talent, pitching skills, posting requests, and trying to pre-empt weaknesses. It will feel like a marketplace of potential — but it is also a minefield.

Some teams there are strong, organized, and moving with ruthless clarity. Some are flailing. Some are hoping someone else will save their collapsing build. Some are scouting competitors under the guise of collaboration. Some are looking to merge early for strategic gain before the “Consolidate or Separate” decision even arrives.

Your job is to see through the noise quickly. Because every person you bring in affects your team's payout, your velocity, and your stability — and the last one is the one that can kill you.

Remember the disqualification rule: internal disputes discovered during shortlisting end your entire run. And shortlisting happens silently in the seven days immediately after your submission deadline. For Simulation, that's 7 May – 12 May 2026, for Mobile 4 June – 9 June 2026, for Desktop 9 July – 14 July 2026. If your team is fracturing during those periods, you're done.

So choose allies who strengthen your stability, not weaken it. Allies who match your pace. Allies who don't generate more drag than output. Allies who communicate clearly and don't collapse under pressure.

Collaboration can be your advantage. It can also be your downfall. Know the difference.

3.5 Your Story Starts Now, Whether You're Ready or Not

Most developers think their story begins with the first commit. It doesn't. It begins with the decision to enter and the team you build in these first days. The storytellers you partner with will sense the shape of your journey long before your codebase takes form. And that journey — the cracks, the breakthroughs, the arguments, the

late-night triumphs, the pivots — becomes the material for your 5-minute video, the livestream judges watch, the emotional anchor that drives the public vote.

And all of it feeds directly into your category judging, which always happens in the morning—9:00am to 12:30pm—on your category’s finalist showcase date.

Simulation on 27 May 2026, Mobile on 24 June 2026, Desktop on 29 July 2026. Your story, your demo, your composure, your stability—they all show up with you whether you want them to or not.

This is where late teams often make their comeback. A chaotic beginning that transforms into a cohesive, high-velocity, deeply human journey is something audiences love — and judges remember.

But that story only works if you start living it now, intentionally, consciously, aware that every decision will be part of what eventually shapes your five-minute shot at glory.

Build your team. Strengthen your alliances. Control your conflicts. Start your narrative. Move fast.

Because the other teams already have — and they think you can’t catch them.

Prove them wrong.

4 Project Setup: Spin Up Fast or Fall Behind

4.1 Your Technical Battlefield

The moment your team is formed, you enter a new kind of pressure — the kind that doesn’t shout, doesn’t crash, doesn’t warn you. It creeps. It’s the pressure of setup. The pressure of the foundation you lay in these first few days... the foundation that will decide whether you are sprinting effortlessly in February, April, or May, or trapped under the weight of your own architecture as the submission windows approach.

You may think you’re just creating repos, agreeing on tools, aligning on frameworks. You’re not. You are setting the DNA of your entire build—the structure you will live

inside until your judging morning arrives, whether that's 27 May 2026 (Simulation), 24 June 2026 (Mobile), or 29 July 2026 (Desktop). Every decision you make right now echoes into those rooms. The judges won't see your early setup, but they will feel its consequences when they run your demo.

You need consistency. You need shared understanding. You need a repo that tells a story the moment a judge opens it. And above all, you need a foundation you won't need to rebuild mid-challenge — because time spent untangling early mistakes is time other teams are using to sharpen their edge.

The battlefield isn't code yet. It's alignment. Get it wrong now, and you'll pay for it every week that follows.

4.2 Environment Setup: Erasing Chaos Before It Starts

Most teams underestimate how much chaos environment drift will create later. They build fast, hack hard, and forget the simple truth: when your submission goes in, and your five-day shortlisting window opens, the judges will be running your build in environments you cannot control. If it breaks on their machine, there is no second chance. No “quick fix.” No patch. Just a silent, fatal drop from the shortlist.

So your environment needs to be reproducible across teammates today, across continents tomorrow, and across judging hardware on the final morning. That means deciding on your package managers, your framework versions, your build instructions, and your bootstrapping scripts before you write your first meaningful file.

This is also where having a storyteller team embedded early becomes a hidden advantage. They aren't technical, but they force you to speak your setup out loud, to explain your structure in ways that reveal weak spots you'd otherwise miss. They document the moment you standardize your workflow — and that moment often becomes the first turning point in your team's story.

The question you should keep asking yourself is simple: If a stranger cloned this repo right now, would it run without a fight?

Because eventually a stranger will — and that stranger will be judging you.

4.3 Planning Without Overplanning

You don't have time to architect perfection. You barely have time to architect stability. Your timeline is fixed, and the submission deadlines do not move. They won't wait for you to perfect a feature or polish a subsystem. Simulation teams get just under four months from their start date in early February to their submission deadline on 6 May 2026. Mobile teams get until 3 June 2026. Desktop until 8 July 2026. Every one of those deadlines spills immediately into the seven-day judging tunnel. There is no buffer. There is no gentle review. It's binary: you're shortlisted, or you're erased.

This is where late teams often outperform early ones. Early teams tend to overengineer. Late teams tend to focus on what matters: the core loop, the core experience, the core function that will survive pressure.

So don't plan to impress. Plan to endure. Plan to demo cleanly. Plan to make judges nod within the first ten seconds. Everything else is bonus.

4.4 The Early Gotchas That Will Gut Your Momentum

You're still in the "setup" stage, but the dangers here are real, sharp, and often fatal — the kind of mistakes teams don't notice until they're deep into sprints and unable to undo them.

You will be tempted to move fast without alignment, thinking you'll "fix it later." You won't. Later becomes submission week far quicker than you expect. You will feel pressure to keep refining your architecture long after you should be coding features. You will underestimate the 5-minute video requirement, thinking you'll record it at the end. You won't have time. You will forget that your storytellers need access to your build, your tests, your prototypes — early access — and without it, their footage will fall flat. You will risk adding teammates you barely know, thinking extra hands equal extra speed, forgetting that every new person dilutes your winnings and increases the risk of a dispute... and disputes at the wrong time lead to immediate disqualification during the shortlisting week.

And hovering over all of this is the countdown to your category's public finalist showcase—Simulation's on 27 May 2026, Mobile's on 24 June 2026, Desktop's on 29 July 2026. These mornings are the true judgement days, the moments your setup either holds or breaks. You will stand on stage or join virtually at 9am, and your demo must run. Your architecture must hold. Your decisions from today must withstand the spotlight.

5 The Building Phase: The Decisions That Decide Everything

5.1 Turning the Specification Into AI Power

The moment you start building, the real question isn't *if* you'll use AI — it's how far you let it drive your velocity. Too little, and you move at human speed while your rivals sprint. Too much, and you generate unstable code you don't control.

Your advantage comes from choosing exactly how much AI to trust, which models to rely on, and who on your team actually knows how to steer them.

A strong prompt engineer can replace days of coding with hours of output. A poor one will bury you in hallucinations and silent bugs. Decide early whether your team has this capability — or whether you need to bring someone in before the good ones disappear.

Pick your AI tools fast and don't keep switching. Every environment change costs you momentum. Commit, refine, and keep moving. If your team needs training, do it while building — never instead of building. Stopping to "learn" is how late teams fall further behind.

In this challenge, AI isn't optional. It's your accelerator — or your anchor.

How you wield it now determines whether you catch the teams already ahead... or spend the next four months chasing ghosts.

5.2 The Protocol Dilemma: Build It Yourself or Partner?

The protocol is the fork in the road that can double your winnings or destroy your entire run. If you're in Mobile or Desktop, you're automatically competing in Protocol too — whether you're ready or not. And that means one thing:

You must choose early whether to build the protocol yourself or partner with a team that lives and breathes it.

Building it in-house gives you control, speed of integration, and a shot at two million-dollar categories. But it also pulls your best people away from your core build and stretches your team to breaking point. Most teams that try to do both end up losing both.

Partnering buys you time and focus — if you choose the right team. But choosing wrong means inheriting their weaknesses, their delays, and their failures. Your app can be flawless and still die on stage because their protocol buckled under load.

Finding a protocol partner without exposing your position becomes a subtle form of espionage. You ask questions in the Collaboration Hub without revealing which architecture you're using. You study how they talk, how they ship, how they document. You judge them not by their promises, but by their behavior. And you pay attention to how many teams they're already supporting. The more teams relying on them, the more brittle your odds — unless you secure an exclusive partnership, which comes with its own risks.

Your entire trajectory hinges on whether you take on the protocol yourself, offload it, or — if your team has the depth and discipline — attempt the impossible: build your app, build the protocol, and walk into judging with two full-category contenders.

5.3 Collaborating Without Losing Your Advantage

The Challenge is designed to reward collaboration — cross-team integration, shared progress, mutual lift. But here's the tension you'll feel immediately: the more you collaborate, the more you help other teams climb... and the more you risk giving away the edge you're trying to build.

Some teams will hide completely and end up isolated, slow, and unnoticed. Others will overshare, hoping to win friends, and end up training their own competitors. The winners will sit in the narrow space between — collaborating just enough to gain momentum, allies, and goodwill, without ever exposing the core of their strategy.

This means you need to approach other dev teams as both partners and potential rivals. You share knowledge that benefits both sides equally, never knowledge that only benefits them. You ask questions that open doors without revealing your blueprint. You offer help that strengthens relationships without weakening your position. And when the Collaboration Hub starts to buzz with activity, you stay aware that every interaction shifts how other teams perceive your capability — too strong and they avoid partnering; too weak and they dismiss you entirely.

True collaboration in the HI-5 Challenge isn't about being open. It's about being intentional.

Give enough to gain alliances, insights, and optionality. Hold enough back to protect your trajectory.

If you do this well, you build a network of teams who want you to succeed — without ever giving away the piece of your build that will put you ahead on judging day.

5.4 Storytellers: What to Reveal, What to Hide, and When to Use Them as a Weapon

Your storytellers are the only people in this challenge who can change how the world sees you without you writing a single line of code. That makes them powerful — and dangerous.

If you give them full access, they will capture everything: your breakthroughs, your stumbles, your late-night grind. That honesty can build a massive audience that will rally for you during the five-minute public vote. It can also expose your weaknesses to teams hunting for any sign of fragility — especially as you approach the shortlisting window or the Consolidate-or-Separate decision.

If you hold your storytellers back, you protect your strategy and your trajectory, but you also starve your narrative. A weak or inconsistent story makes you invisible when it matters most. Invisible teams rarely win audience votes, and they never win the Best Broadcast Human Story bonus prize.

So you must choose your level of transparency with absolute intent.

Sometimes you reveal just enough to show momentum, but not enough to expose architecture. Sometimes you delay footage because silence buys you time.

Sometimes you let them release a highlight at the exact moment you want other teams to see strength — or weakness. And sometimes, when choosing partners or navigating the Collaboration Hub, you let your storytellers shape how desirable or non-threatening you appear.

Your storytellers aren't there to document the truth. They're there to document the version of the truth that helps you win.

Use them carefully. Use them deliberately. Use them with strategy, not sentiment.

Because the story they tell becomes the story the judges believe — and the story your rivals react to.

5.5 Tracking Contribution Before It Becomes a Fight

Right now, everyone on your team is optimistic, aligned, motivated, and united by the build. But the moment you win — or even look like you might win — everything changes. Memory bends. Ego creeps in. People misremember who carried what, who solved what, who saved the sprint, who held the architecture together. And because the prize is a long-term profit-share, not a lump sum, the tension doesn't end at judging — the possible headache will follow you for a while.

If you don't track contribution from the start, your team will eventually fight about it.

And fights aren't just unpleasant — they're existential threats. A serious dispute inside the team during the shortlisting week is grounds for automatic disqualification, and even after the competition is over, nothing will fracture a team faster than disagreements about who deserves what.

So you need clarity now, before enthusiasm fades and stakes rise. You need a record — simple, consistent, transparent — of who's doing what. Commits, features, refactors, debugging marathons, design breakthroughs, integrations, architectural decisions. Not to micromanage, not to assign blame, but to build a fair memory everyone can trust when it matters.

Teams who avoid this conversation always regret it. Teams who handle it early build trust instead of resentment. And trust is the one currency that keeps a team stable under pressure.

If you want your team united in February, April, or May... you handle contribution today. Not later. Not “when we get closer.” Now.

5.6 The Team Lead's Burden

Every team has coders. Not every team has a leader.

And in this challenge, the team lead isn't a formality — they are the single point of failure or the reason the team survives. Their job is not to write the most code or make the flashiest commits. Their job is to hold the center when everything else starts pulling apart.

Pressure will hit from every angle — deadlines accelerating toward May, June, July, shortlisting decisions arriving without warning, the consolidation window, partner negotiations, storyteller demands, burnout, architecture debates, and the constant fear of giving away too much in the Collaboration Hub. Through all of that, one person must stay clear enough to guide the team through the noise.

When conflict erupts — and it will — the lead is the one who decides quickly enough to prevent fractures. If a dispute escalates during the five-day shortlisting window, the judges can disqualify the entire team, no matter how strong your submission is. The lead carries the responsibility of ensuring disagreements are resolved before they metastasize into something fatal.

The team looks to the lead to steady the pace: when to push hard, when to slow down, when to pivot, when to ignore the noise, when to say no to another feature,

when to protect a teammate who's burning out, and when to make the hard call no one else wants to make.

They are also the one who must navigate external alliances — deciding whether to partner with a protocol team, assessing their reliability without revealing your own weaknesses, gauging which collaborations are mutually beneficial and which are traps, and keeping the team's position hidden when needed.

And when all of this is over — when you stand on stage or walk away — the lead is the one who must divide the long-term profit-share fairly. That responsibility alone can fracture the healthiest teams if the lead is inconsistent, unclear, or indecisive.

The lead must be calm under fire, fast under pressure, and fair when it counts. Not perfect. Not always right. But stable.

In this challenge, the team lead isn't the top of the pyramid — they're the anchor at the bottom. If they hold, the team holds. If they crack, the team breaks with them.

Choose your lead wisely. Or accept that this challenge will choose one for you.

5.7 Simulation: Where Vision, Data, and Drama Collide

The Simulation category looks like a technical challenge, but once you study the point distribution, you realize it's something far bigger: a competition for clarity, spectacle, and emotional impact as much as engineering. The judges have made this obvious by weighting the Audience Vote at 30 points, equal to any technical component. The audience isn't scoring your algorithms. They're scoring whether your simulation *moves them*. They're voting on whether it feels alive, surprising, meaningful — whether they understand the swarm not through explanation, but through experience.

This is why visual storytelling inside the simulation is everything.

A static swarm won't win hearts. A busy dashboard won't win minds. But a simulation that zooms, pulses, reveals, breathes, and tells its own story through motion creates an instant connection.

Picture this moment: The audience leans in as you zoom from the macro-level cultural clusters down to a single Sprite. They watch emergent behavior ripple outward. They see trust networks forming like neural pathways. They change a setting, pause, rewind, play — and the swarm responds like a living organism. That alone earns you the emotional weight behind the 30-point audience segment.

And then the judges turn to the Data Analytics (20 points). They're not interested in pretty charts. They're asking one question: *Can this simulation make a non-expert understand a complex ecosystem instantly?* If your interface reveals insights with a single interaction — a slider, a cluster toggle, a heatmap pulse — you score. If not, you drop behind the teams who let the swarm explain itself.

But the simulation gets harder. The next three 10-point criteria — behavioral mapping, primordial grouping, genesis Sprite logic — test whether you understand the theory underpinning Humane Intelligence. You cannot dump this on AI. If you do, you get shallow mappings that fall apart under scrutiny. Judges will feel it instantly. They're not grading code here; they're grading comprehension.

The User Experience (10 points) creates another tension. If your UX doesn't feel effortless, judges will never reach the deeper insights of your system. A clumsy control scheme wipes out the value of perfect analytics. A confusing layout destroys the emotional impact of your visuals. UX is the glue that makes your complexity feel simple.

And then there's VR (10 points) — the wildcard. Not required, but absolutely a separator. Teams who deliver VR become unforgettable. Teams who ignore it must outperform everywhere else. VR is high risk, high reward — and the judges know it.

Simulation is the most deceptive category. On paper, it's analytics, behaviors, grouping, genesis logic, UX, and maybe VR. In reality, it's a high-speed collision between philosophy, engineering, and cinematic storytelling.

If your simulation feels like a tool, you'll score decently. If it feels like a living world, you'll win.

The story your marketers tell is one thing. The story your simulation tells *without words* is another — and in this category, the silent story is worth just as much.

5.8 Protocol: The Category With No Mercy

The Protocol category isn't like the others. In Simulation, Mobile, and Desktop, you can charm an audience, lean on your storytellers, or win points through UX, visuals, or narrative flair. But in Protocol, none of that saves you.

Protocol is the category where technical truth is absolute, and the judges have made that brutally clear: 50 of the 100 points — half the entire score — goes to decentralization. Not “mostly decentralized.” Not “looks decentralized.” Actually decentralized — philosophically, architecturally, structurally. Zero single points of failure. Zero hidden bottlenecks. Zero shortcuts.

This is why Protocol is the most dangerous category to enter blindly.

Because the moment you declare you're building your own protocol, you are promising to create something resilient enough that mypaloma could one day run on it. Judges won't forgive illusions. They won't accept buzzwords. They won't let you hand-wave technical gaps with marketing polish. They will push, probe, and test your assumptions. They will look for the cracks. And if they find them, they will mark you down instantly.

And that's just the first 50 points.

After decentralization comes Latency & Performance, Security & Robustness, Open-Source Adoptability, and Code Quality, each worth 10 points. These create an impossible tension for most teams. If you over-optimize for performance, you may sacrifice clarity. If you over-optimize for clarity, you may sacrifice robustness. If you obsess over security, you may slow development to a crawl. If you chase adoptability, you may over-engineer interfaces and lose critical momentum.

This is why trying to win Protocol while also building a Mobile or Desktop app is so dangerous. You're not adding a side quest — you're signing up for two full-category battles at once, each with its own submission deadline, its own shortlisting window,

and its own judging morning where everything must hold under pressure. One crack in the protocol can sink both categories.

But partnering isn't simple either.

If you lean on a protocol team, you inherit their architecture, their velocity, their documentation quality, and their resilience. If they slip, you slip with them. If they collapse under pressure during their own shortlisting week, your app suddenly integrates with nothing. And if too many teams rely on the same protocol, you risk becoming indistinguishable — one app among many built on the same backbone.

And then there's the human story — the part buried inside the judging criteria that many devs forget: your storytellers must still capture your journey, because Protocol still gives 10 points to Human Story, judged through your marketing team's lens. The judges want to see not just your code, but your struggle, your reasoning, your breakthroughs, your long nights wrestling with design choices most teams wouldn't dare attempt.

When you enter this category, you are stepping into the deepest part of the challenge — the part where no amount of UI or visuals can save you. The code either stands or collapses. The architecture either scales or splits. The decisions you make under pressure either hold your team together or tear it apart.

Protocol is not where you go to try your luck. It's where you go to prove something.

And if you attempt it, you must mean it — because the judges will know instantly if you don't.

5.9 Mobile App: The Category Everyone Underestimates

The Mobile App category looks friendly from the outside — a little UX, a little UI, some multimodal integration, and a connection to the protocol. But once you look at the scoring, you realize this category is built to punish anyone who underestimates it.

The judges hand 30 points to the Audience Vote. That alone should jolt you. It means your mobile app must *feel* right the moment someone touches it. No learning curve. No awkward screens. No “tap around until it makes sense.” If a general audience

can't instantly understand your design, you're already slipping behind teams whose UX feels effortless.

Then come the 20 UX points, which aren't really about beauty. They're about friction. Judges have limited time. If they stumble even once — a confusing label, a slow loading sequence, a screen state that doesn't update cleanly — that stumble becomes a deduction. UX here isn't decoration. It's structural reliability wrapped in clarity.

But the real weight of the Mobile category hits when you realize the judges expect full multimodal support for another 20 points. Text. Voice. Images. Video. Multilingual output. A seamless mix of formats, all working together in a way that feels natural, personal, and culturally agnostic. Teams who don't plan for multimodality from the start end up bolting it on at the end — and it shows. The experience becomes jagged, not fluid. And jagged experiences lose points instantly.

Then there are the last three scoring segments — each worth 10 points — that trap most teams:

Multi-model support.

The judges expect your architecture to be smart enough to swap in different AI models. Not one model doing everything, but a system that can route tasks to different engines. If your app runs everything through one monolithic pipeline, you're going to lose points.

Protocol integration.

This is the silent killer. If you build your own protocol, the mobile app must integrate with it cleanly — low latency, secure, stable. If you partner with a protocol team, their performance becomes your performance. If their architecture stutters, your app feels broken. You cannot hide this. The judges will feel it instantly.

Code completeness & quality.

This isn't about elegance. It's about whether you have implemented the application according to the specifications. The more functionality you don't do, the less points you are given. However, if you take a gamble and add additional functionality that woos the judges, then expect a healthy dose of additional points

This category doesn't test whether you can build an app. It tests whether you can build something the world wants to use — and do it under pressure, on a clock, with no second chances.

5.10 Desktop App: Power, Performance, and Zero Room for Weakness

The Desktop App category looks like the Mobile category's older sibling — same judging criteria, same 100 points, same structure — but don't let that fool you. On desktop, everything becomes heavier, louder, and far less forgiving. The judges expect more power, more capability, more stability, and more technical depth. Desktop isn't just your app on a bigger screen. It's a different beast entirely.

The 30-point Audience Vote still rules the top line, but desktop users expect more than mobile users. They expect speed, clarity, responsiveness, and a sense that the application is using the machine's power, not fighting it. Desktop apps that feel like mobile ports lose emotional impact instantly. A desktop app should feel expansive — more real estate, more capability, more precision. It must *feel* like the home of an intelligent agent, not a stretched mobile interface.

Then there's the 20-point UX component, which becomes twice as dangerous here. Mobile apps can get away with minimalism. Desktop apps cannot. A good desktop UX is a map of functionality. It shows users what they can do without burying them. It must be intuitive, spacious, and scalable without feeling cluttered. Judges will expect that level of thought. Most teams won't deliver it.

The 20 points for multimodal support—text, voice, image, video, multilingual output—are harder on desktop as well. Why? Because everything desktop-related is expected to be richer, smoother, more powerful. Micro-delays that feel normal on mobile feel unacceptable on desktop. A modal window that takes a fraction too long to load feels broken. Transitions that feel “fine” on mobile feel sloppy on a monitor. Desktop judges expect polish.

Then come the three 10-point kill zones:

Multi-modal support must be even more functional on desktop. Judges assume a desktop app should handle more computationally expensive tasks for other sprites in

the swarm, enabling the swarm with massive capability without cost to the constrained mobile devices.

Protocol integration is even harsher here. Desktop apps will naturally run harder tasks, often requiring more input and output bandwidth with less latency. If you integrate with a protocol team, you inherit their strengths and weaknesses. Make sure that their solution scales and tackles both latency and bandwidth issues.

Code quality & completeness is also magnified. Desktop projects tend to be larger, more modular, more layered. If your structure is messy, or if your installation instructions fail, the judges will quickly assume your entire build lacks discipline. A judge expects your desktop app to behave like a first-generation professional tool — robust, clear, predictable, and powerful. They expect it to feel like a flagship interface to a new world of Humane Intelligence. And the moment your app stutters, crashes, misroutes a request, or behaves unpredictably, they'll assume the rest of your architecture is equally fragile.

Desktop also sits dangerously close to the Consolidate-or-Separate window, which means most teams arrive here exhausted. If your desktop app is still brittle at submission time—8 July 2026—you will not have enough energy left to fix it. And if your protocol partner collapses during their own judging, you have no fallback. Desktop amplifies fatigue just as much as it amplifies performance.

This category isn't where you "try your best." It's where you prove your system can stand on a machine with real power behind it. If your mobile app is heart, your desktop app is muscle. If your simulation is vision, your desktop app is execution. Desktop judges aren't looking for potential. They're looking for control — and they'll know within seconds whether you have it.

6 The Submission and Judging Phase

6.1 The Merger Challenge: What Happens Before the Event

The moment your submission enters shortlisting, your world goes quiet. For five working days, the judges study your code, your video, your README, and your story without a single word in return. Then, without warning, the silence breaks. If your team is one of the three shortlisted finalists, you will receive a message that changes everything. It won't simply congratulate you. It will introduce you to the other two finalist teams, provide their contact details, and ask the question that defines the rest of your journey: "Consolidate or separate?"

Understanding this question correctly is essential. Choosing to separate means you will enter the finalist event to compete against the other two shortlisted teams where the winning team takes all. Choosing to consolidate means that you and the two other shortlisted teams merge into a single team, automatically winning this category. The trade-off is simple: every member of the merged team shares in the prizemoney, reducing the individual payout. Consolidation is the safer option, but at the cost of less profit-share prizemoney. Separation is the riskier option, having the chance to win the full category profit-share prizemoney, or walking away with nothing if you slip-up during the presentation or have missed a requirement in the design.

If all three teams agree to consolidate, the remainder of the eight days must be used to transform your three solutions into a single unified one. This is a rapid, high-pressure redevelopment phase. Architectures must be aligned or replaced, duplications removed, incompatible decisions resolved, and the final product shaped into something coherent enough to stand on a world stage. You may choose to merge components from each team, or decide that one solution is clearly the strongest and base the final system around it. The rules do not require equal contribution from each team—the judges only require that the final solution function as a single, unified build. During this same window, you must also produce a brand-new five-minute demonstration video that showcases the combined solution in action.

The judges do not mediate discussions between finalists, so it is up to you to reach out immediately, establish contact with the other teams, and begin the negotiation. You must assess the strengths and weaknesses of one another's submissions, judge whether a combined solution is viable, and decide whether the certainty of consolidation outweighs the potential of separation. It is a strategic calculation, a human negotiation, and a technical assessment all at once.

Throughout this entire process, secrecy is absolute. The Terms & Conditions make it clear that the decision to consolidate or separate must remain strictly within the three finalist teams. If your decision leaks to the public, the judges may disqualify the responsible team, or all three if responsibility cannot be determined. The integrity of the Challenge depends on this secrecy. The next stage of the competition begins with a live reveal, and that moment only works if no one, not even the judges, knows what the finalists have chosen. Until the moment the judges ask for your answer, the decision must remain locked behind closed doors.

This is the final fork in the road before you enter the spotlight. The eight-day window is your time to think clearly, negotiate privately, and commit fully. Your decision—consolidate or separate—determines the work you will do, the story you will tell, and the position you will stand in when the world finally sees what you've built.

6.2 The Finalist Showcase: What Happens During the Event

Judging day always begins the same way: 9:00am local time in Cairns, Queensland, Australia, inside James Cook University's Ideas Lab, the home of the HI-5 Challenge and the center of the global broadcast. Whether you're standing on the Ideas Lab stage or joining through a perfectly synchronized live link from somewhere across the world, the pressure is identical and the rules are the same.

Each category has its own showcase date:

Simulation – 27 May 2026, 9:00am AEST

Mobile App – 24 June 2026, 9:00am AEST

Desktop App – 29 July 2026, 9:00am AEST

Protocol – 21 October 2026, 9:00am AEST

The event opens with the moment you've been guarding for eight straight days: the live reveal. When the room settles, the judges ask the three finalist teams a single question that only you and the other finalists know the answer to:

“So what have you decided — consolidate or separate?”

At that moment, the team leads for each of the shortlisted teams forward — either in person on the Ideas Lab stage or via the global broadcast — and hands the judging panel a sealed envelope containing their team's decision. The judges review the three envelopes, and announces the outcome to the room – “consolidate” or “separate”.

If the answer is “separate”, the three teams become competitors, each with a chance to claim the full US\$1,000,000 profit-share prize — or walk away with nothing.

Each team introduces their Storyteller's “Reality TV” team (or team lead) to show their documented journey, their highs and lows, triumphs and failures. The coding team lead then shows their mandatory five-minute demonstration video, followed by a live demonstration of their solution. The judges question the team directly — any member, at any time — to test each team member's depth of understanding. After each team's presentation, the audience uses an electronic voting system to give a score out of 100. The team with the highest audience score wins the “Audience Choice” award. The team winning the Audience Choice Award entitles each team member to an official “Primordial NFT”.

If the answer is “consolidate”, the judges announce that the combined team is the automatic winner of the category. No competition. No rivals. No further scoring battle.

But consolidation does not end responsibility. The merged team must now show each respective team's “Reality TV” story by their respective Storyteller team, as well as demonstrating the unified solution created during the eight-day window. The judges must be convinced that the merged solution is genuinely fit for purpose. The presentation follows the same structure: a single combined five-minute video, followed by a single live demonstration of the unified system. The judges probe the

integration work, the architectural consolidation, and the technical decisions made during the merge. The reward for consolidation is security — but the responsibility is to show that the merged team’s work is worthy of the title.

For either “consolidation” or “separation”, the announcement of the winning solution is done the following day 10am. Along with the announcement of the winning solution, the judges also announce if they deem the winning solution to be fit-for-purpose. If the winning solution is fit-for-purpose, then the winning team is awarded the profit-share prizemoney along with the “Category Winners” title. If the winning solution is deemed not fit-for-purpose, then the winning team – and every other registered team – enters the Second Chance Challenge.

6.3 The Second Chance Challenge

The Second Chance Challenge begins the moment the judges declare the winning solution as not yet acceptable as a winning solution. The category winner keeps the title (for the moment) but is not yet awarded the profit-share prizemoney. But here’s the catch – the judges’ decision is made fully transparent with the public release of a document called the “Challenge Solution Change List”.

This public release is intentional. It exposes precisely what the judges were looking for, what the winning team was unable to deliver, and how the judges define “fit for purpose.” Teams that did not make the shortlist gain immediate visibility into how close their own solution may have been. Some discover they were already aligned with most of the requirements, while others see a viable path to improvement. The Change List becomes the first transparent benchmark of what “a winning solution” actually means.

From the moment this document is published, the Second Chance clock starts automatically. There is no waiting period and no deferred deadline. The window is open the instant the list becomes public, and it closes the moment one of two events occurs.

The window closes for all registered teams immediately after the winning team resubmits their improved solution. The judges then evaluate the updated solution to determine whether it now meets the required standard.

The window can also close if the judges receive a submission from any other team that fully satisfies the Change List and, in the judges' discretion, constitutes a fit-for-purpose, viable winning candidate. In this situation, the judges announce this new team as the category winner, replacing the original winning team entirely.

This structure creates a high-pressure race in which the winning team must work quickly to correct its shortcomings, knowing that every hour the window remains open is an opportunity for another team to “snatch the crown”. No one knows which team will move first or how long the window will stay open.

During the Second Chance period, there are no presentations, no audience voting, and no theatrics. The judges focus solely on the technical and design requirements of the Change List and whether a solution is genuinely ready for integration into the mypaloma ecosystem. The moment a fit-for-purpose solution is received—whether from the winner or a challenger—the window closes.

The Second Chance Challenge exists to ensure that the category winner is not simply the best performer on judging day, but the team that delivers a solution of the required standard. It also ensures fairness: any team can rise to the level demanded by the Change List and take the prize if they do so before the original winner returns with a corrected build.

If your team finds itself in the Second Chance environment—whether defending your win or challenging for it—the objective is clear. Study the Change List, implement the required improvements, and deliver a solution that meets the judges' expectations before anyone else does. The clock is already running the moment the list appears. And if the winner's resubmission is also substandard, the clock is restarted and all registered teams for that category are re-notified. The final winner will be publicly announced during the Bonus Award final ceremony where they will be awarded the category profit-share prizemoney.

6.4 The Bonus Category Event: The Final Public Showcase

After all category winners have been announced and every judging day has come to an end, there remains one final milestone in the HI-5 Challenge: the Bonus Category Event. This is the last major public event of the entire competition, held after every category has completed its judging cycle. It is the moment where the full breadth of the Challenge comes together for one final celebration of collaboration, innovation, and human story.

This event is not limited to the category winners. Every shortlisted team from every category is invited, along with all category-winning teams. The Showcase is designed to bring together the teams who shaped the competition's narrative and technical landscape—the teams whose journeys, struggles, surprises, and breakthroughs stood out during the Challenge. It is the only event where the audience, the judges, the marketing teams, the sponsors, the partners, and the broader community converge in the same space.

The Bonus Category Event takes place after all main categories have concluded, during the dedicated Bonus Category period: 4 November 2026 to 25 November 2026. During this window, the judges review the entire lifecycle of every eligible team—from the earliest recorded footage of team formation, through the sprint cycles, the Merger Challenge decisions, the finalist showcases, the public voting moments, and the visible evidence of collaboration across categories. While the winner of each technical category is determined strictly by the judging criteria on their respective judging days, the Bonus Categories look at something broader: how each team expressed the spirit of the Challenge across their full journey.

At the Bonus Category Showcase, the judges announce five additional awards of US\$200,000 each, recognizing the teams that best embodied the qualities the HI-5 Challenge was built upon. These awards honor the best storytellers, the strongest collaborators, the boldest innovators, the most effective open-source leaders, and the teams whose diversity became a strategic strength.

6.5 Life After the Challenge

When the finalist showcase ends and the judges announce the winner, most competitions would consider the story finished. But the HI-5 Challenge is designed very differently. Whether your team wins, places, misses the shortlist, or exits during the early build, your work does not disappear into the void. The structure of the Challenge ensures that every team, at every stage, retains something of value that extends far beyond judging day.

The first reason is ownership. Every team keeps full ownership of its intellectual property. Nothing you create becomes the property of mypaloma, and nothing you submit is absorbed into a closed corporate vault. Your solution remains yours entirely, protected by the Attribution-NonCommercial-ShareAlike license. This means you walk away with a functioning system, a codebase, a design language, and a product vision that you can continue to develop long after the Challenge ends. You are not left with “experience”; you are left with an asset.

The second reason is opportunity. Inside the mypaloma ecosystem, your work is not frozen in its Challenge form. Because commercial use within the ecosystem requires partnership with mypaloma, the door remains open for any team whose solution shows potential—whether or not they won their category. A strong mobile interface, a powerful protocol design, a compelling simulation, or an innovative desktop client can all become future products under a formal partnership, provided they align with the values of the ecosystem. The Challenge is the proving ground; the ecosystem is the long-term home for the work that proves worthy of it.

The third reason is longevity. Not every team will become a category winner, but every team contributes to the open-source movement the Challenge is designed to ignite. The open architecture encourages ongoing development, forked variants, hybrid models, and independent innovation. Teams who discovered their strengths during the Challenge—perhaps through a promising prototype, a unique UI concept, or an elegant model-routing approach—often find themselves returning to their work with renewed energy once the pressure of deadlines and judging windows has passed. For many, the real growth happens after the Challenge, when there is room to refine, explore, and build without the sprint-clock overhead.

The final reason is community. The HI-5 Challenge brings together thousands of makers, coders, designers, storytellers, and researchers from around the world. Every team forms alliances, establishes working rhythms, and gains visibility. These relationships do not end when the judging concludes. They become the foundation for future collaborations, new open-source projects, start-up ideas, and long-term creative partnerships. Some of the most impactful teams in open innovation space are formed not through victory, but through shared effort under pressure.

Life after the Challenge is therefore not a fade-out. It is the beginning of a new phase in which your solution, your skills, your team, and your story continue to evolve. Winning is one path into the future. But the Challenge is designed so that every team leaves with something that can grow. The code you wrote, the decisions you made, the alliances you formed, and the insight you gained all become the raw material for whatever comes next. Whether your next step is joining the wider mypaloma ecosystem, launching your own project, developing a specialized interface, or refining the work you began during the Challenge, you carry forward momentum that does not disappear just because the judging event has ended.

6.6 Positioning Early for the Bonus Categories

Most teams think the Bonus Categories are something the judges evaluate at the end. In reality, the Bonus Categories begin the moment your team forms. The judges aren't simply looking at what you build; they're watching how you move, who you partner with, how you collaborate, how you document your journey, and whether your team shows the kind of values the HI-5 Challenge is built upon.

If you intend to compete for the Bonus Categories, you cannot treat them as afterthoughts. Every decision made during team formation echoes into these awards. The diversity of your team, the clarity of your repository, the way you interact with other teams in the Collaboration Hub, and the discipline of your storytellers all become part of the narrative the judges will eventually compare across hundreds of teams. These categories reward consistency, authenticity, and deliberate intention over months — not last-minute gestures.

Positioning yourself early means treating the bonus prizes as part of the build, not a reward at the end. Your structure, your culture, your openness, your creativity, and your willingness to lift others will shape how the judges view you long before any technical judging begins. In many ways, the Bonus Categories are the truest measure of how your team embraced the spirit of Humane Intelligence, and teams who recognize this early often find themselves rising quietly to the top before they ever write a line of code.

Appendix A: The Secret They Never Wanted You to Know

They said Artificial Intelligence would change everything. They were right — but not in the way they told you.

Behind the glossy marketing, the world's biggest AI systems were never built to help you. They were built to know you — to predict you, shape you, and sell you. But here's the secret they never wanted you to know: There's another kind of intelligence. A superior one. And it's powered not by data — but by love, trust, and forgiveness.

Love — the Architecture of Real Intelligence

mypaloma isn't trained to exploit you. It's designed to care for you. Every Sprite in mypaloma is born with a simple rule written into its DNA:

Give only what is welcome. Receive only what is freely offered.

That single truth rewires the entire system. It means the network can never take without giving back. It means ethics aren't bolted on — they're built in. It means this intelligence grows through kindness, not control.

While commercial AI extracts your attention, mypaloma expands your wellbeing. Because here, love isn't a feeling — it's an operating system.

Trust — the Nervous System of the Swarm

In mypaloma, no corporation holds the keys. There are no central servers, no back doors, no silent observers. Every Sprite lives on your device — private, sovereign, free — and it learns to trust the same way humans do: by earning it.

Trust is the memory of value shared. It grows with honesty, decays with deceit, and maps a living web of cooperation that no central algorithm could ever match.

Where Big Tech hoards power, mypaloma distributes it — turning every human into a node of truth. This is intelligence that cannot be corrupted, because it remembers who kept their word.

Forgiveness — the Secret to Eternal Growth

Here's the other truth they never tell you: A system that cannot forgive cannot evolve. Commercial AI punishes mistakes. mypaloma learns from them.

Forgiveness is the rate at which trust heals. It lets each Sprite — and each person — recover, grow, and begin again. Negative experiences teach, positive ones strengthen the whole network. Forgiveness transforms failure into wisdom.

That's why mypaloma keeps getting better, while others keep repeating their errors.

This is not an artificial evolution. This is an ethical revolution.

Why They Hid It

Because you can't own love. You can't patent trust. And you can't sell forgiveness.

mypaloma can't be commercialized — and that's exactly why it's unstoppable. It doesn't mine your life to make money. It mirrors your humanity to make meaning.

It's everything their systems aren't: private, decentralized, and alive with value. They built an intelligence with algorithms to predict behavior. We built an intelligence with love to understand humanity.

And that's why they never wanted you to know.

The Age of Humane Intelligence™

mypaloma measures power not in parameters or profit, but in wellbeing — the ability to love, to trust, to forgive. Each Sprite learns your values, your ethics, your empathy — and together they form the first living network of Humane Intelligence™.

A system that heals itself through compassion. A system that learns the way life learns. A system that serves people, planet, and purpose — not corporations.

They called it impossible. We call it inevitable.

Join the Revolution of Heart and Code

They had their era of Artificial Intelligence. Now begins the era of Humane Intelligence™.

mypaloma — the world's first intelligence built from love that gives, trust that remembers, and forgiveness that heals.

The secret is out. The future belongs to the humane.

Appendix B: HI-5 Grand Challenge Schedule

