

This document is part of the **Frontiers Design Documentation**, a set written to teach, not to instruct. Each entry takes one subsystem, explains what that subsystem is in the wider language of tabletop design, surveys how games have solved it, and then shows what Frontiers does as one worked answer among many.

It is written for several readers at once:

- Designers using Frontiers who want to understand why it works the way it does.
- Designers using Frontiers who want to change it, and need to know what a change will cost.
- Designers building something else entirely, who want the design space mapped before they choose.

Frontiers is a toolkit, and a toolkit is only useful if you can see what each tool is for. None of this is a rule; the rules live in the Rulesets, and what follows is the reasoning, the history, and the alternatives.

There will be terminology here that is unfamiliar to casual players. Treat this documentation as advanced content, written for designers interested in building. It will not teach you every term or phrase it uses, but it will do its best to define them as they arrive.

---

## What are Characters

---

A character is anyone in the game who is made of the game's rules: the people the players run, and the people the GM runs. That covers a lot of ground, from a fully realized protagonist to a guard with three relevant numbers, and one of the first things a system decides is whether all of those are built from the same parts.

Some systems build everyone on one chassis, so a villain is assembled exactly like a hero, just with different choices. Others give the GM a separate, lighter framework, so the things the players fight are quick to make and quick to run, at the cost of never quite matching how a player character works. Where a system lands says something about its world: whether everyone in it is made of the same stuff, or whether the players are the real characters and everyone else is scenery with statistics.

Underneath that, nearly every system has the same two phases for the characters it does fully build:

1. **Assembly.** Building the character in the first place, from setting numbers to choosing who they are.
2. **Advancement.** Changing the character over time, and what growth even means in this system.

This document covers those two, because they are the parts every designer has to answer and the parts a player recognizes immediately. It also keeps one eye on the player-versus-non-player

question throughout, because in most systems the difference between the two comes down to how much of these two phases each one gets.

## Assembly: the first choices set the tone

Even the very first step of building a character is a design decision with consequences. Take one question: are a character's stats **rolled** or **chosen**?

Call of Cthulhu rolls them. You generate an investigator partly at random, and part of the game's texture is that you are handed a person to inhabit instead of a person you optimized, which suits a game about ordinary people caught in something larger. A point-buy or array system does the opposite: you design the character you want, deliberately, which suits a game about capable protagonists built for a purpose.

Neither answer is the correct one. Rolled stats produce characters you discover, and chosen stats produce characters you design, and that single fork changes what creation feels like and what kind of character walks out of it, and it is only the first of many such forks. Every assembly step, order of operations, how identity is layered on, how much the fiction is involved, is a lever on the same question: how much of this character is authored, and how much is arrived at?

There is a further question hanging over all of it: **how much does your game care about roleplay?** A system built for character-driven play front-loads identity, background, and relationships, and makes them mechanically real. A system built for tactical or survival play may treat those as flavour a player adds themselves, and spend its creation rules on capability instead. Neither is wrong, but assembly is where a game tells a player which kind it is, before a word of setting has been read.

## Advancement: what growth means here

The second phase is how a character changes over time, and it is where a system reveals what it thinks a character is for.

The common shapes are worth naming plainly, because a designer picks among them early and lives with the choice:

- **Levels.** Growth arrives in discrete tiers, each a bundle of improvements. Legible and satisfying, and it makes power a clear staircase, at the cost of flexibility, since a level is a package, not a single choice.
- **Points.** Growth is bought incrementally, a character spends earned points on whatever they like. Flexible and granular, at the cost of bookkeeping and the risk that players optimize toward a few best purchases.
- **Milestones.** Growth happens when the story reaches a marked moment, instead of from tracked accumulation. It ties advancement to narrative and removes the accounting, at the cost of the steady mechanical drip some players want.
- **Use-based.** A character improves the things they actually do. Intuitive and character-driven, at the cost that players will use their best tools to grow them and neglect the rest.

The hard part of advancement is **balancing and scaling** whichever shape you choose. Whatever the method, a designer has to answer how fast growth comes, how much each step is worth, and whether a character at the top of the curve is still playing the same game as one at the bottom.

Grow too fast and the numbers outrun the content, too slow and play stagnates. Steps that are too large make early choices stop mattering, and steps that are too small stop advancement feeling like anything. These are the questions that actually decide whether a progression system feels good, and they are the same regardless of which shape you picked.

How is a character built, and what does it mean for them to get better?

---

## The Approaches

---

Because a character subsystem does two jobs, it is worth looking at the two separately. How games assemble a character and how they advance one are different questions, and a system can be conventional at one and adventurous at the other.

### How games assemble

The assembly question is really about how much a character is authored versus arrived at, and how many layers of identity the system stacks.

#### ROLLED AND OCCUPATION-DRIVEN

Call of Cthulhu generates much of an investigator at random, then hangs their skills off an occupation. You are handed a person as much as you build one, which fits a game about ordinary people pulled into something beyond them. The rolled element is tonal: you inhabit who you got, instead of optimizing who you wanted.

#### LAYERED SELECTION

The 5e SRD lineage stacks identity in tiers, a species, a background, a class, often a subclass, each contributing a piece. This is the most familiar shape, and its strength is expressive range: the combinations are many, and each layer is a recognizable handle. The cost is a longer creation process and more interlocking parts to learn before a first character exists.

#### THE DESCRIPTIVE SENTENCE

Cypher compresses identity into a single spoken statement: I am a [descriptor] [type] who [focus]. Three words, chosen from lists, and the character is defined, an adjective for flavour, a noun for their core role, a verb for what they do best. It is fast, genre-portable, and unusually evocative for how little it asks, at the cost of the fine granularity a layered system gives.

#### PLAYBOOK-FIRST

Thirsty Sword Lesbians and the wider Apocalypse World tradition hand a player a playbook, a complete character archetype with its moves, its stats, and often its emotional shape already

framed. Creation is fast and produces a three-dimensional character almost immediately, because the playbook front-loads not just capability but who this person is and how they relate to the others. The tradeoff is that you are working within a curated archetype rather than building from parts.

## **How games advance**

Advancement is where a system shows what it thinks a character is for, and the shapes vary more than assembly does.

### **TIERED, IN DISCRETE STEPS**

Cypher builds a tier out of several separately-bought improvements, a stat increase, an ability, a skill, an edge, each purchased before the character reaches the next tier. Growth is a series of small chosen things instead of one bundle, and notably it treats advancement as a representation of the story rather than the point of play. This is the model that most directly inspired Frontiers' own two-track split, and the part I took from it is the insistence that growth arrive as several small chosen things instead of one lump.

### **LEVEL PACKAGES**

The traditional lineage grants a bundle of improvements at each level, sometimes chosen, sometimes fixed. It is legible and satisfying, a clear staircase of power, but it concentrates growth into rare, lumpy moments, and when the package is identical every level, advancement becomes automatic instead of a decision.

### **EXPERIENCE INTO PURCHASES**

Point-buy advancement lets a character spend earned currency on whatever they like, incrementally. Flexible and granular, at the cost of bookkeeping and a tendency for players to converge on a few optimal buys.

### **ROLEPLAY-TIED**

Games like *Thirsty Sword Lesbians* mark experience through play, hitting emotional or narrative beats instead of defeating challenges, and cash it into advances. Growth follows the story's shape, which suits a character-driven game and gives up the steady mechanical drip some players want.

### **LITTLE OR NONE**

At the far end, some games barely advance characters at all. *Cairn* has no levels and no experience track; a character grows mainly through what they find, learn, and carry, and stays mechanically close to where they started. This keeps every character fragile and every fight meaningful for the whole life of a campaign, and it suits short, lethal, exploration-driven play. It gives up the long arc of getting stronger, which is exactly the point for the games that choose it.

The two axes work independently, which is the useful thing to take away from all this. A game can assemble characters from a fast descriptive sentence and then advance them through a rich tiered system, or hand out elaborate layered builds that barely change afterward. When you design a

character subsystem, you are answering both questions, and you are free to be bold on one and conventional on the other.

---

## What Frontiers Does

---

Now, with that overview to catch you up to speed, here is the approach Frontiers takes, and the reasoning that led there.

Frontiers assembles a character from two identity layers over the shared chassis, and advances them on two tracks:

- **Origin** and **Archetype**, the two identity layers, over the Attributes, Skills, Abilities, and Equipment the other documents cover.
- **Marks**, small personal growth earned often, from an **Individual Pursuit**.
- **Levels**, large shared growth earned at story beats, from the **Party Pursuit**.

Full rules live in the Rulesets; what follows is why the shape looks like this.

### Two identity layers: nature and nurture

Most systems split identity into three or four layers: a species, a background, a class, often a subclass. Frontiers uses two, **Origin** and **Archetype**, and the split is nature against nurture.

Your **Origin** is what you were born with and where you come from, your nature and your past. Your **Archetype** is what you were trained into, what you have made yourself, your nurture. Between them they answer the two questions that actually distinguish characters, and they do it without the four-tier stack that most systems carry.

Two layers rather than four is the same call the engine keeps making. The species/background/class/subclass breakdown is common, detailed, and more than a chassis needs, so Frontiers takes the smallest frame that still captures the real distinction and leaves the elaboration to the system. A designer who wants species and background as separate things can split Origin in two; a designer who wants sub-specializations can layer them under Archetype. The engine gives you the clean pair and expects you to build on it.

### Creation follows a logical flow

The recommended order runs: concept, then Origin, then Archetype, then Attributes, then Derivatives, then Skills, then Abilities and Equipment, then Pursuit.

The order runs as a throughline, and you should depart from it wherever your game needs to, since each step exists to give you what the next one needs. Concept tells you who you are aiming at. Origin and Archetype tell you what that person should be good at, so when you reach Attributes you are choosing with a target in mind instead of guessing. Attributes then produce Derivatives, Skills and gear flesh out capability, and Pursuit caps the build with the character's reason to act. A

designer can reorder freely, but this sequence front-loads the information, so nothing is chosen blind.

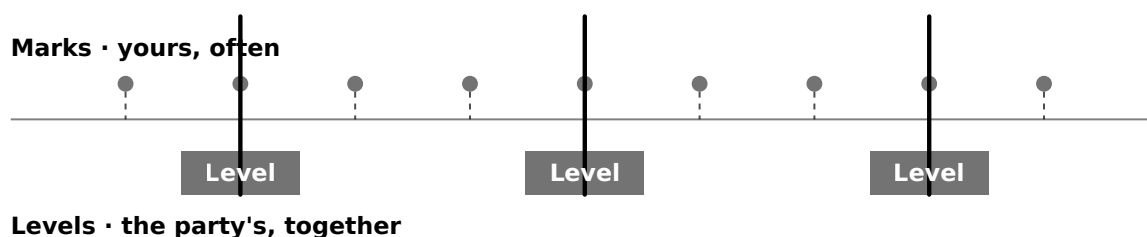
On the numbers, Frontiers recommends a **chosen** spread over rolled stats, because a chosen array is easier to balance and produces characters built to a purpose. Rolled generation is offered alongside it, because some games want the character you are handed and not the one you designed, and there is no reason not to support both.

## Two-track advancement

This is the part Frontiers drives hardest for. Advancement runs on two tracks at once:

| Track              | Comes from                    | Grants                                          |
|--------------------|-------------------------------|-------------------------------------------------|
| Individual Pursuit | One character's personal goal | <b>Marks:</b> small, immediate, personal growth |
| Party Pursuit      | The group's shared goal       | <b>Levels:</b> the larger beat, gained together |

Two tracks, running at different speeds



Marks arrive from your own Pursuit, roughly one a session. Levels arrive for everyone at once.

The split is a reaction to how most systems handle growth, which is to blend everything into the level-up. Advancement arrives rarely and all at once, so play spends long stretches feeling static until a big level lands and dumps a pile of changes. The common alternative is worse: hand out the identical package every level, so growth is guaranteed but never a choice. Neither feels good. One makes growth lumpy and sparse; the other makes it automatic and flat.

Frontiers separates the two textures. **Marks** are the small, frequent, personal growth, a bump to two Attributes, a new Skill, an Ability, earned whenever a character meaningfully advances their own Pursuit, roughly once a session. **Levels** are the large, shared beat, gained by the whole party together when the Party Pursuit resolves.

Something is therefore almost always arriving, and the big jumps stay distinct from the small mechanical gains instead of being mashed into one event. The lineage here is Cypher, whose advancement breaks a tier into several separately-bought steps for much the same reason: growth should be a series of small chosen things, not a single lump.

Two design choices keep the tracks honest. Marks are **capped at three per Level** with no bank and no carryover, so a character cannot hoard personal growth and outrun the table. And Levels are **not earned by Mark count**: the party rises together when the shared story turns, not when

someone fills a bar. No character out-levels the group, and the whole table always has a reason to care about the same goal.

There is a consequence worth naming, because a player will find it. Three Marks a Level across ten Levels is thirty Marks, and an Attribute Mark raises two Scores by one each, so an all-Attribute character would generate +60 Score. A character built on the recommended spread starts 4 Rank-points above baseline across the four Attributes, and the default ceiling sits 40 Rank-points above it, leaving 36 points of headroom. (On the roll-over scale that is 44 against a ceiling of 80; on the d100 it is 220 against 400, which converts to exactly the same 36.) The ceiling therefore binds after eighteen Marks, and a player who plans to spend everything on Attributes will run out of room somewhere around two-thirds of the way through a campaign.

That is intended. It forces diversification without a rule that says "diversify," and it keeps the other Mark options, Skills, Abilities, Resilience, from competing against an Attribute track that could absorb everything. But it is a wall, and a player who plans an all-Attribute build should be able to see it coming, not discover it at Level 7. A designer raising the Attribute ceiling, lengthening the campaign past Level 10, or changing the Mark cap should recheck this interaction, since all three move where the wall sits.

## **Pursuits drive growth, and the table authors them**

Both tracks are powered by **Pursuits**, goals a character or party is bending toward, because tying advancement to pursued goals is a broad, genre-agnostic engine that drops into almost any game. Growth comes from chasing what you want, which works as well for a heist crew as for a knightly order.

The engine does not legislate who writes a Pursuit. In practice a Party Pursuit tends to be GM-set, since it functions much like a milestone, the goal the campaign is currently bending toward. An Individual Pursuit is more personal, and where it comes from is left open, because a character's private goal is a matter of roleplay and discovery. The engine hands you the two tracks and trusts the table to decide whose goals fill them.

For tables that do not want the narrative lens, plain milestone and experience-point advancement are both fully supported, and everything downstream, the Mark menu, Capacity, Archetype features, works unchanged under either.

## **NPCs use the same chassis**

Frontiers ships no dedicated NPC generator, because NPCs are content and depend entirely on the system built on top. But the character rules are designed so that making one is trivial: an NPC is a character with the advancement layer left off.

In practice an NPC comes out as **Origin, Archetype, Attributes, and done**. Everything past that, Skills, Abilities, Equipment, even a Pursuit, is available if the moment calls for it and skipped if it does not. A nameless guard needs three numbers; a recurring rival can be built out as fully as a player character.

The same chassis produces both, which means a GM never learns a second system to populate their world, and an NPC that becomes important can be fleshed out rather than rebuilt. Advancement is the line between a player character and most NPCs: the PC carries the two tracks, and the NPC usually does not.

## **Is this the best solution for Characters?**

Character building is the assembly point for everything else in the engine, so the aim was to keep it light where it could be and structured where it mattered. Two identity layers instead of four, a creation order that informs itself, and an advancement system that gives players something often, not rarely.

The two-track split is the piece *Frontiers* is most committed to, and the piece most open to a designer's own spin. Origin and Archetype can be resized, the Pursuit lens can be swapped for milestones or experience, and the whole thing is built to have a system's own identity content laid over it. The structure is the opinion. What fills it is yours.

There is a good chance you want a different shape for identity or advancement. The final section below sets out variant and alternative rules to use in their place, along with what each one changes and what it will cost you elsewhere.

---

## **Variants**

---

Variants keep the philosophy intact. Two identity layers, two advancement tracks, and Pursuits driving growth. What changes is where the dials sit. Mix as suits your game.

### **Tune the identity layers**

The default is two layers, Origin and Archetype, nature and nurture.

#### **SPLIT A LAYER**

Break Origin into a separate species and background, or add a sub-specialization under Archetype, for a game that wants more expressive range at creation. This is the four-tier stack the default deliberately compressed, so add it only where the extra granularity earns the extra choices.

#### **COLLAPSE TO ONE**

Merge Origin and Archetype into a single identity pick, for a lean game where a character is one clean archetype and nothing more. Fast, at the cost of the nature-versus-nurture distinction.

### **Tune the Marks**

Marks are the personal track: capped at three per Level, spent from a fixed menu.

## CHANGE THE CAP

More Marks per Level makes personal growth faster and looser; fewer makes it precious. The cap exists to stop a character outrunning the table, so moving it up widens how far ahead one player can get between Levels.

## CHANGE THE MENU

Add, remove, or reprice what a Mark can buy. This is the cleanest place to inject a system's own identity into advancement, since the menu is where a Mark becomes something specific.

## CHANGE THE TEMPO

The recommended roughly-one-Mark-a-session is a pacing default. Move it to suit your table. Faster or slower shifts how granular growth feels without touching anything structural.

## ADD A CATCH-UP VALVE

Marks are granted at GM discretion, which is what makes them a record of whose story is getting attention. It also means a distracted or conflict-averse GM can produce an uneven party without anyone deciding to. The engine leaves this to the table by default, on the grounds that an uneven spread is information rather than a bug, but a designer who would rather not rely on that has three cheap options:

- **A floor.** A character more than a set number of Marks behind the party earns one automatically at the next Level.
- **A catch-up rate.** A lagging character's Pursuit grants Marks at double rate until they are level with the group.
- **A Level grant.** Everyone receives one Mark at each Level regardless of Pursuit, so nobody advances at zero.

Each preserves the diagnostic, since Marks still show who is getting screen time, while removing the failure mode where a quiet player simply falls behind and nothing in the rules notices.

## Tune the Pursuit lens

Growth is driven by Pursuits, but the lens is swappable.

## SWAP TO MILESTONE OR EXPERIENCE

The engine already supports plain milestone advancement, where the GM grants Marks and Levels at story beats, and experience points, where a currency buys them at fixed prices. Both run on the same downstream machinery; only the trigger changes.

## CHANGE WHO AUTHORS PURSUITS

Formalise what the default leaves open, players always write their own Individual Pursuits, or the GM sets all of them, or the table drafts them together. The engine declines to legislate this, but your game can.

## Alternatives

---

Alternatives replace a phase rather than tune it. One caution first, and it applies to every entry here: **Characters is the assembly point for the whole engine, so almost any change to it reaches into other subsystems.** Advancement feeds Capacity, + HP, and Archetype features; identity feeds what a character starts with; creation order touches how every other system is introduced. None of these swaps is fully self-contained, so trace each one outward before committing.

### Rolled assembly

Make rolled stats the default instead of the chosen spread. A character is generated partly at random, and the player builds outward from what they got.

This is a tonal change more than a mechanical one. Rolled characters are discovered rather than designed, which suits a grittier or more capricious game, and it costs the balance that a chosen array guarantees, some characters will simply come out stronger than others. Nothing downstream breaks, since Attributes feed the same Derivatives either way, but the game's feel shifts at the root.

### Single-track advancement

Collapse Marks and Levels into one track. Growth arrives in one texture instead of two, whether that is a single leveling ladder or a single stream of small gains.

Simpler to run, and it gives up the thing the two-track split was built for: the sense of getting something often while big beats stay distinct. Watch the downstream effects, since Levels carry Capacity growth, + HP, and Archetype features. Fold everything into one track and you have to decide which beats now carry those, or the character's foundation stops scaling correctly.

### Levelless

Cut advancement almost entirely, in the manner of Cairn. A character grows through what they find, learn, and carry and not through Marks and Levels, and stays mechanically close to where they began.

This keeps every character fragile and every conflict meaningful for a whole campaign, and it suits short, lethal, exploration-driven play. It is also, notably, close to how Frontiers already builds NPCs, characters without the advancement layer, so the engine is well set up for it. The large cost is downstream: Capacity, + HP, and Energy scaling all ride on Levels, so a levelless game has to either fix those at their starting values or find another thing to grow them. Growth through gear and fiction has to carry the weight that Levels usually would.

## **Layered identity**

Go the other way from the default and adopt the full species, background, class, and subclass stack, for a game that wants deep, combinatorial character builds.

This buys expressive range and a long, familiar creation process, at the cost of the speed and low overhead the two-layer default was chosen for. It is mostly self-contained, since it changes what a character starts with rather than how the engine runs, but it does lengthen every character's on-ramp, which is a real cost for a game that wants players making characters quickly or often.

Whichever you choose, remember that this is the subsystem everything else is assembled from. A change to identity or advancement is rarely a change to Characters alone, so the honest first step is always to ask what else moves when this does.