JUMP POINT

ISSUE: 11 01

JUMP POINT

ISSUE: 11 01

IN THIS ISSUE →→→

FROM THE COCKPIT

GREETINGS, CITIZENS!

Welcome to February's **Jump Point**! It's currently all hands on deck in the Persistent Test Universe, with thousands of you putting the upcoming Alpha 3.18 patch through its paces. Thank you to everyone that's played and fed back the experience via the Issue Council – this data is vital to the devs tirelessly working to get the wealth of new content out to the Live servers.

It goes without saying that this patch is proving to be a tough one to implement, and it's all down to one ground-breaking feature and the topic of this issue's interview… Persistent Entity Streaming (PES). Lying on the development road between Object Container Streaming and Server Meshing, PES is a huge hurdle to be cleared in achieving the ultimate goal of a seamless and evolving playable universe. To find out everything we could about this cutting-edge technology, we spoke to one of the main devs behind its development, Chief Technical Officer Benoit Beausejour.

We're then taking a detailed look into the PU's latest career, Salvage, including what it is, how it works, and how it was developed.

As usual, we have an enthralling new lore piece from the Narrative team; this time from their newest writer, Jeremy Melloul.

Finally, we're showcasing the 'verse's fastest and most agile racing ships in playing card form. Are our stats community-approved? We doubt it, so let us know where we went wrong on Spectrum.

Thanks again to everyone who reads **Jump Point**, and don't be shy about letting us know if there's anything you want to see in a future issue.

We'll see you in the 'verse,

Jump Point Team

JumpPoint@cloudimperiumgames.com

Writer & Copy Editor : Martin Driver **Narrative Team Writers:** Adam Wieser, Jeremy Melloul, & Cherie Heiberg
Layout Design: Michael Alder **Imagery:** Simon Ravenhill, Sarah Longley, & Luke Heath
Marketing Producer: Simon Jennings **Narrative Team Producer:** Blythe Gumminger

PERSISTENT ENTITY STREAMING THE DEFINITIVE EXPLANATION

Alpha 3.18 is *Star Citizen's* biggest patch to date, with significant developments to the engine and general make-up of the universe.

This vast backend update is the second in a trio of cutting-edge technologies being developed in pursuit of a universe that evolves with the players that inhabit it. The first, Alpha 3.3's Object Container Streaming, delivered the ability to create a seamless expansive universe at a scale never seen before. Now, Persistent Entity Streaming adds the groundwork for ever-existing items and the visible impact of habitation and exploration, and will ultimately lead to a single playable space for all via Server Meshing.

Naturally, Persistent Entity Streaming was a significant and hugely complex undertaking. So, in pursuit of the definitive explanation of this groundbreaking technology, we spoke to one of the key developers behind it.

NAME:	BENOIT BEAUSEJOUR
ROLE:	CHIEF TECHNICAL OFFICER
TEAM:	TURBULENT
LOCATION:	MONTREAL, CANADA

JUMP POINT: Hi, Benoit! Thank you so much for talking to us. Before we get started on Persistent Entity Streaming (PES), could you give us a brief explanation of your role and what you've worked on for *Star Citizen*?

BENOIT BEAUSEJOUR: *I am the CTO at Turbulent and have technical and creative oversight on all web platform projects and teams as well on the Online Services teams that develop the next-generation backend for Star Citizen. I have been on the project since the very beginning, dating back to 2012, where we built the crowdfunding platform for the game.*

More recently, our online teams got tasked to help the PES teams develop a persistence backend to store the game data.

JP: Here we go... what does PES do in the Persistent Universe (PU)?

BB: *PES is a game system that persists entities. The game servers simulate the game world and update thousands of parameters on thousands of entities in as much real-time as possible. The goal of PES is to take all this data (that normally only lives in the memory of the game server and gets wiped at each restart) and store it efficiently and safely in databases that can be used to reload the state of the game universe exactly as it was. So, PES is currently in charge of the full persistence of the game world so that actions of the players affect the world in permanent ways.*

JP: Why is PES such an important feature?

BB: *Persisting the game world by default is important because it gives players more agency in modifying the world they play in. As they move around in a PES-enabled world, they will encounter objects in a state modified by previous events. Whether it's the remnants of a space battle or a water-bottle totem built by a player in Grim HEX, PES gives life and meaning to the world in ways that are normally lost in non-persistent games where the state of the world is reset periodically.*

JP: What will it enable or lead to in the future?

BB: *We are still discovering all the ways in which PES will benefit the project but we anticipate that many game features will become "persistent by default," which will make their development much easier and standardized. Having full persistence on entities, including dynamic variables, means that developers and designers can come up with new objects, features, and missions that leverage persistence to achieve a gameplay effect without having to design their own persistence.*

We already see in the game the effect that PES has on gameplay, as players use persistence to give life to the world, either by littering or by cleaning it up. Overall, it is changing our approach to game design.

JP: So... how does PES work?

BB: *PES is a giant data pipeline composed primarily of the game server, the Replication Layer, and the Entity Graph Service that handles the Directed Acyclic Graph (a very large graph).*

The game servers have a representation of the game world in memory inside the Replication Layer; picture this as a tree of objects with properties that change in constant time. As those changes occur, the servers have open outgoing data streams to the Entity Graph and transform those changes into mutations for the Entity Graph Service and send them over at a high rate.

The Entity Graph Service then enqueues these mutations in a persistent queue and subsequently processes the messages to apply them to a graph database. All those writes end up modifying a persistent version of the graph that is stored inside datafiles within the database as collections of nodes and edges. We are currently using SST files for their property of handling high write rates.

This then allows the Replication Layer to reload its entire state by querying the Entity Graph by performing streaming tree traversals on boot, restoring the state of the game world for players.

Technically, each "shard" has its own database and state, providing an alternate reality of the same game world. The data is completely formed as a graph through and through (list of nodes and edges), so it is natively understood by all components.

JP: How does PES relate to Object Container Streaming and Server Meshing?

BB: *PES is part of the Server Meshing project in that it provides the persistent state needed to split the simulation across servers. In a Server Meshing world, instead of loading the entire world, game servers will load only the "territories" they are assigned to from the Entity Graph by changing the root at which they perform the traversal.*

This allows any game server (or replicant in a fully formed mesh) to grab state and authority over a territory by loading the data from the Entity Graph.

Object Container Streaming enables all this as it is the basis of the data format.

JP: What were the major challenges with implementing PES?

BB: *Developing PES was challenging in that the data rates at which the servers submit their updates are very high. This means that concurrency handling and resiliency must be perfectly handled. For example, handling where a part of the tree must be moved out while another component is trying, at the same time, to write to that subtree. We needed to come up with clever APIs and algorithms to solve eventual consistency problems as well as message ordering. As mentioned before, PES is a giant data pipeline, so the messages need to flow but also notify the sender when jobs are performed (as markers). This means that every component in the chain is critical to a point.*

We spent a lot of time ensuring the resiliency of all the different components to make sure all kinds of failure scenarios were handled. On such large-scale deployments, things are known to break in the most mysterious ways!

Data size is also a major challenge when designing a system like PES. Since we need to store millions if not trillions of entities, the database design of read-and-write queries and API must consider how large certain areas are. For example, the game world is split into zones (which are given nodes in the tree). As players add entities to the world, "hot zones" can form, making traversal costly in that area. Dealing with data size at every step, from the size of ships to entity counts, was a major challenge and required optimized data structures at every step.

One large challenge is also designing APIs for game systems that have a very large footprint and complexity. Part of PES also includes inventory system persistence, which uses the same data format but stores player-assigned data inside the global database. We refer to "stowed" entities as entities that you put into an inventory. For example, storing your ship at an ASOP terminal effectively stows the ship in your inventory at that location. In those events, the ship root and its children are moved to the global database for cold storage. "Unstowing," on the other hand, brings an entity tree back into the game world on a shard.

This system design, and the associated API design related to long-term persistence and entitlements, was very challenging, as there are many undocumented use cases of the previous inventory system that came into play with a lot of ties into how insurance works. Those data flows are complicated to design for because data moves from one database to another, with support for partial wipes and post-patch restoration, which can either open up game exploits or prevent normal gameplay from functioning.

JP: What's next for PES?

BB: *For PES, it's performance optimizations, bug fixing, and monitoring. We are still discovering issues that stem from players using the feature and we fully expect to have to update, change, hack, saw, and modify the system to enable players to do the crazy stuff they love doing!*

JP: Following PES, what's the next major technical undertaking for *Star Citizen*?

BB: *Server Meshing is a big technical hurdle that will rely on the stability and performance of PES to achieve the results we all want: large shards with many players that persist and that provide meaningful gameplay experiences.*

JP: What are you and your team working on now?

BB: *The Online Services team are now supporting the Network team in making Server Meshing by providing orchestration and control support for shards. As the game server and Replication Layer and its components become truly distributed (running as separate processes, scaling independently), there is a need for processes that understand what a shard needs to operate, report its health, and perform corrective actions and interactions with the cloud infrastructure to keep it running. We're moving onto that mission!*

So there we are, the definitive explanation of Persistent Entity Streaming from its foremost authority! A huge thank you to Benoit Beausejour for this fascinating dive into this ground-breaking feature and the technology surrounding it.

BEHIND THE SCENES:
SALVAGE

Thanks to Persistent Entity Streaming, Alpha 3.18 will introduce an all-new and much-anticipated career. One of the initial crowdfunding goals, Salvage was unveiled with a grand plan:

Salvage Mechanic: *Salvage isn't an aside, it's a career with its own mechanic, story tie-ins, and universe-shaping endgames. Search the galaxy for a host of valuable and interesting secrets using both the flight and FPS components. Discover the secrets of the ancient Hadesians, locate valuable components and cargo... or go down in history as the first to make contact with an entirely new alien race!*

While uncovering the mysteries of the universe's earliest inhabitants is likely a long way away, the debut of Salvage will give players the means to make a more peaceful living in the stars than the well-established combat-focused careers. And in line with the goal set out over a decade ago, it'll offer both first-person and vehicle-focused gameplay.

To celebrate the upcoming implementation, we're looking into its launch state and talking to two of the devs behind its development.

DAWN OF SALVAGE

At its core, Salvage is about removing and recycling the hulls of incapacitated ships. Whether the player sells the recovered material for profit or uses it is up to them, but the first part of the journey is removing metal, known as hull scraping. Requiring a dedicated salvage laser, hulls can be scraped by hand using the Greycat Pyro Multi-Tool or via dedicated ship.

As a ship's hull is scraped, Recycled Material Composite (RMC) is produced. This valuable material can be hauled to a landing zone to be sold for profit or used to patch up a damaged (but not destroyed) ship to help it limp to a rest stop for proper repair.

In Alpha 3.18, there are three ways to produce the all-important RMC:

1. DRAKE VULTURE

New in Alpha 3.18, Drake's Vulture is the latest in the brand's departure from make-do pirate favorites to effective career ships for somewhat respectable pilots. A single-minded industrial ship built to scrape and process derelict hulls, the Vulture will also support 'munching' when it's added in a subsequent patch.

2. GREYCAT PYRO MULTI-TOOL (W/ CAMBIO-LITE SRT)

The ubiquitous Greycat Pyro Multi-Tool takes on a new role in the ripping apart and repairing of damaged ships thanks to the Cambio-Lite SRT module. Although naturally lacking the capabilities of a dedicated salvage platform, the Multi-Tool is ideal for quick cut-and-shut jobs or exploring the career path without going all in on an industrial ship.

3. AEGIS RECLAIMER

The PU's first salvage vehicle has been flyable since early 2018, but the latest patch finally gives it the utility that defines it. Built for long-range Salvage trips into the depths of space, this titan of industry is the most effective way to break ships apart and is well-prepared for future releases with unmanned salvage drones and ore processing.

SHIPS & SALVAGE

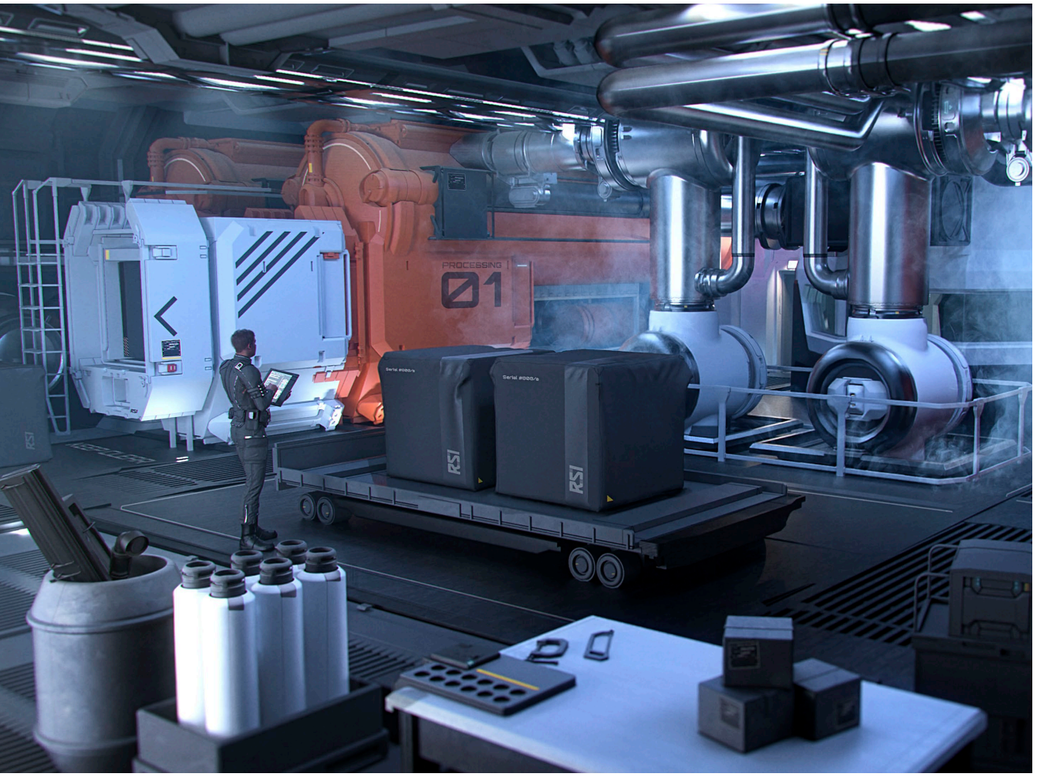
We caught up with the UK Ships team to find out how the Salvage career impacted the development of *Star Citizen*'s vehicles. Senior Vehicle System Designer Geoffrey Coffin explains how the mechanic affects older vehicles and the ships currently in development.

"Our work with ship breakup has thankfully helped to a degree with the salvaging process, so we already have breakup plans for every ship we make. But now that the hull can be scraped off from ship sections, how the exterior paneling looks from the inside has become more of a consideration in places. Sometimes, when salvaging a wing, you'd be able to see clear through the wing, for example."

So, the detailed process of ship creation meant that most vehicles were already set up for the process of being broken down. The same goes for the materials each vehicle is constructed from.

"Thankfully, with the salvage process, the properties of the section of the ship you're harvesting are taken from the assigned material. So sheet metal, plastic trim, etc. will all have their own unique values, and this is something the system handles automatically."

The PU's existing salvage vehicles are currently being put through their paces. With the continued development of the wider career, what does the future hold for Salvage vehicles?

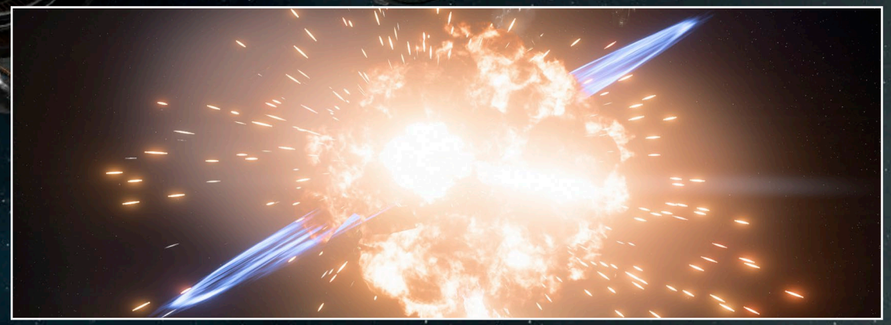


"We currently don't have any other salvage ships on the horizon. The Vulture was set up to be an entry-level starter ship; something accessible and usable. The Reclaimer has its lasers set up for the initial release of salvage but is still awaiting the cutting and munching gameplay when the tech progresses a little further."

With Salvage also encompassing Repair and Refining in the future, could some of the 'verse's lesser-discussed concepts return to the limelight?

"For Repair gameplay, we have the Vulcan, which is planned to offer Repair gameplay with its drones in addition to being able to re-arm and refuel. We also have the Crucible, which will be a dedicated repair ship with its own landing pad and repair drones. For Refining, we have the Expanse, which is a single-seater and poised to be our entry into the career, and the Galaxy, a larger multi-crew ship (if equipped with the right module)."

So, while no new vehicles are currently planned for Salvage and its related careers, there are several existing ships awaiting production as the gameplay and tech behind them progresses.



SOFT DEATH

Alpha 3.18 sees the introduction of the Soft Death mechanic. Previously, when a ship's hull was irrecoverably damaged, it would explode. However, due to the needs of Salvage, Soft Death was developed to leave ships in a 'scrapable' state, giving scavengers or the ship's crew the opportunity to make something from the loss.

Due to the patch's long residence in the Persistent Test Universe (PTU), Soft Death benefited from significant iteration and testing. Initially, upon reaching the previous threshold for destruction, each ship rolled the dice on a 70% chance of remaining immobile but intact. However, feedback led to the designers changing this to a more consistent requirement to give players the option of seeking Soft Death rather than all-out destruction. As of Alpha 3.18, a ship is guaranteed Soft Death if a model-specific selection of key components, such as the engine and thrusters, are destroyed. For example, take out a Drake Cutter's powerplant through accurate Gatling fire and it'll be left defenseless but scrapable.

However, once a ship's 'soft dead,' it's not plain sailing to a pay day – the crew could still be onboard, armed, and looking for revenge.

DEVELOPMENT, IMPLEMENTATION & THE FUTURE

While most development teams at least touch on each career as it's implemented into the PU, Salvage was pushed over the line by PU Features. We spoke to Lead Systems Designer Thorsten Leimann to find out how Salvage was created.

Leimann explains that the ultimate aim of Salvage is to completely dismantle disabled ships or ground vehicles, starting with removing reusable or repairable components.

"For salvage itself, we want you to take a ship or vehicle fully apart. Let me walk you through our full intended flow:

While the vehicle is still semi-functional, you start by extracting possibly valuable data. The next step is removing all the functional or nonfunctional items that can be repurposed, repaired, or sold.

Once you are done removing all the valuable items from the ship, you can remove the valuable liquids and gasses. This aspect of the Salvage gameplay is called Syphoning.

Once the dangerous fluids and gases are removed, you can start removing the actual hull, which we have in Alpha 3.18. The removal of the hull is key to seeing the points you need to target to proceed to the next phase. The goal here is to bring the ship into a shape that fits into your ship's opening and get it towards the 'grinder' for Cutting or Munching."

The difference between Cutting and Munching is the precision and control over what's processed. For example, the additional control of Cutting allows players to pre-filter certain types of resources for reduced processing. Mulching is the final process, where the unwieldy remains of the target ship are processed as a whole. This output can easily be sold as scrap metal or further processed to potentially generate more profit.

"We want the Salvage loop to tie in well with the entire chain of resource gathering, resource refining, and, in the end, crafting, which will lead back into resource gathering."

The ultimate aim is that the resources gathered by players (via mining, refining, salvage, etc.) make up the various components and items found around the 'verse.

"For example, a ship weapon that, at some point, might make its way back to being grinded down into a resource to processed again."

With the aims locked down and the PU's backend approaching a state to support it, Salvage was put into development in 2022. Leimann details the key considerations when approaching a feature with long-term expectations and contemporary tech restraints.

"There were plenty of considerations, like the minimal viable product that we can release, especially if you look at the whole plan for Salvage, as there are some beats in the Hull Scraping loop that might not be as useful with it 'just' being a standalone feature. Also, there were some technology considerations where we had to align with the teams responsible for the core tech."

Long-term, many salvagers will rely on other careers (and therefore players) to make their living, including explorers to provide lucrative scrap locations, mercenaries for escort into lawless territory, and cargo haulers to shift their product. Though a way away from a vital cog in the ever-turning economy of the 'verse, the first implementation of Salvage is a solid base from which to grow its gameplay and connections to other careers.

"The good thing is that we have quite a good basis to build from, which should make our lives much easier when we get to the next beats of the Salvage gameplay loop. That does not mean we will be able to deliver them faster, since the more we dive deeper into the gameplay the more complex it will get, but it is really satisfying to have taken the first step and started the whole plan."

Salvage's existence relies on Persistent Entity Streaming, so it naturally developed with the feature in mind. However, the ongoing cargo refactor had a significant impact on the new career's production too and caused its own list of issues along the way.

"The biggest impact was the parallel release with the cargo refactor and PES. These were things that from the beginning had to be planned for and even led some of our decisions. For example, because of the physicalized cargo that we have in Alpha 3.18, we decided to introduce the filler station to allow players to see and experience the process of filling a cargo hold. Sadly, we did not have the time to make the process more user friendly, but it is an issue we did not lose track of."

With the first version of Salvage having been iterated on in the Persistent Test Universe (PTU) and the mechanic approaching a Live release, what's next for the EUPU team and the 'verse's latest career path?

"There are three things we have already started working on. On the one side we are starting with the next steps in Salvage, including the Munching gameplay loop, which has the goal of allowing players to clean the 'verse fully of husks, debris, and destroyed ships. Additionally, alongside our tractor-beam work, we are also starting with the item-stripping aspect of the Salvage gameplay loop. On the Repair side, it is something that will, like Salvage, come in waves, though we have one aspect of it in our Engineering gameplay that you will hear about at a later stage."

For players currently getting to grips with ripping hulls apart in the 'verse, this is only the beginning of the wider Salvage and Repair gameplay loop, and it seems like there's plenty more to look forward to in coming patches.

Leimann ends our chat with a shout out to the VFX and Weapons Feature teams, stating that, *"We only took what they prepared and made it work with ships and fit it into our planned Salvage gameplay."*

Thank you to Thorsten Leimann of EUPU team for taking the time to share the process of implementing Salvage.

>> Quality Assurance Team: Testing Salvage

(This is one of the processes used by the QA team to test Salvage. CVars are disabled on all public servers.)

To use the Salvage & Repair fire modes you need the GRIN Multi-Tool with the Salvage & Repair Module and Salvage & Repair Multi-Tool Canister attached.

You should see the Salvage & Repair UI and Battery UI on the Multi-Tool display.

Aiming at a ship should reveal the Salvage & Repair Lens AR Card and update the Multi-Tool UI.

Obtain the Tool

Multi-Tool
1.Equip the Grin Multi-Tool: - *grin_multitool_01*
2.Stow the Module: - *grin_multitool_01_salvage_repair*
3.Attach the Module:
Open the Weapon Attachment menu.
Press J
(OR)
Hold F, trigger Customize interaction on weapon.
4.Pick up the Canister:
- grin_multitool_resource_salvage_repair_01
5. Equip the Canister:
Press R
(OR)
Open the Weapon Attachment menu:
Interact with canister
6.Attach the Salvage & Repair Module

CVar:
g_spawnEntity grin_multitool_01
g_spawnEntity grin_multitool_01_salvage_repair
g_spawnEntity grin_multitool_resource_salvage_repair_01

(OR)
i_giveitem grin_multitool_01_default_salvage

Salvage

Salvage is the default fire mode for the Salvage & Repair Module. Press V to cycle to Salvage fire mode:

To be able to salvage:
- *Aim at a vehicle*
- *Vehicle shields need to be disabled*
- *Be inside the beam's range*
  [Note that the tool will detect a target at a different distance]
- *Canister must NOT be full*
- *Aim at a part of the vehicle hull that has visible hull remaining*

Salvaging should increase the canister ammo and damage of the ship hull (part health and visually).
Reloading should replace the canister with an empty canister.

Repair

Salvage is the default fire mode for the Salvage & Repair Module. Press B to cycle to Repair fire mode:

- *Aim at a vehicle*
- *Vehicle shields needs to be disabled*
- *Be inside the beam's range*
  [Note that the tool will detect a target at a different distance]
- *Canister must NOT be full*
- *Aim at a part of the vehicle hull that does not has visible hull remaining*

Repairing should reduce the canister ammo and partly restore the damaged ship hull (part health and visually).

Reloading should replace the canister with a full canister.

STAR CITIZEN®

TRADING CARDS

—RACING SHIPS EDITION—

MISC RAZOR

TOP SPEED	4
ACCELERATION	4
AGILITY	4
WEAPONS	3
DEFENSE	1

MISC RAZOR EX

TOP SPEED	4
ACCELERATION	3
AGILITY	4
WEAPONS	3
DEFENSE	3

CONSOLIDATED OUTLAND MUSTANG GAMMA

TOP SPEED	3
ACCELERATION	3
AGILITY	3
WEAPONS	1
DEFENSE	3

MISC RAZOR LX

TOP SPEED	5
ACCELERATION	4
AGILITY	4
WEAPONS	2
DEFENSE	5

KRUGER INTERGALACTIC P-72 ARCHIMEDES

TOP SPEED	2
ACCELERATION	2
AGILITY	5
WEAPONS	1
DEFENSE	1

STAR CITIZEN®
TRADING CARDS
—RACING SHIPS EDITION—

VIRGIL LTD

When the Vanduul ‘Kingship’ arrived, the Siege of Tiber that had been ongoing for four years was quickly ended. The UEE forces in the system that had been desperately trying to hold the line were forced to fall back to Virgil.

Unfortunately, the Vanduul were not satisfied with the spoils from a single system. They pursued the routed fleet into Virgil. Without a secondary line to adequately defend the neighboring star system, the UEE could not hold back the raiding clan and the Vanduul unleashed utter devastation, killing millions. In the ashes and tragedy of this disaster lies the meek origins of what is today a universe-spanning personal defense company with a renowned reputation for quality: Virgil Limited.

A WORLD OF OPPORTUNITY

As Humanity spread through the stars, endless Citizens and civilians alike looked to other worlds for a chance to earn their fortunes. In that spirit, many eyes fell on the promising Virgil system after its discovery in 2412. The Selea family was one of many who traveled to the verdant world of Virgil I to take advantage of this opportunity. For many years, they profited from this decision, enjoying a peaceful life working as farmers, passing the work from generation to generation. However, once the Orion system fell to the Vanduul in 2712, life in the Virgil system took on a different tenor. The idyllic Virgil I was transformed into a military support station to supply the UEE forces deployed to guard against the Vanduul invaders in the neighboring systems.

Ponya Selea had grown up eager to join her family working on their farm, but by the time she achieved her Equivalency in 2727, the military-industrial complex had become the world’s main economic driver, surpassing both agriculture and tourism. After graduation, Ponya found a job working in one of Virgil I’s many factories, a large machining plant developing armor plating intended for a variety of imperial defenses.

There, she met her partner, Godri, and together they had a daughter in 2730, who they named Cedra. Ponya’s dedication and natural leadership saw her rewarded for her work and she was soon promoted to floor supervisor within her plant. She’d likely have continued down this path had her life, like every other on Virgil, not been disrupted when the Vanduul broke through Tiber and laid waste to her home.

DAY OF DISASTER

While the Siege of Tiber was ongoing, tensions in the neighboring system of Virgil remained high. There were plans and contingencies in place to account for a possible retreat from the Tiber system. However, when Tiber eventually fell, Virgil was unprepared for the immediacy of the Vanduul onslaught.

In the airspace over Virgil I, the beaten remains of the 81st Battle Group formed a hasty defense to buy time for Squadron 214 to help civilians escape as Vanduul bombers rained waves of explosives onto the surface of the planet.

The machining plant the Selea family worked at was lucky to survive the initial bombing, though not without casualties. It was utter chaos inside the factory, until Ponya managed to rally the surviving factory workers and lead them to their children. Yet they were still far from safe.

With roads destroyed and most transit inoperable, the workers and their children would have to make the trek on foot as Vanduul started to seize control of the planet.

Hiding from the Vanduul ships flying overhead, Ponya led the survivors into the nearby forest of titanic trees. It would be a longer journey, but they hoped the forest's thick canopy would shelter them from danger. They were right... until the Harvesters landed.

They were heard before they were seen, churning everything in their path, from rocks and trees to metal and organic matter. The encroaching shriek of the Vanduul machines made the survivors realize they couldn't outrun them. At least not all of them. Amid the fear and panic, Ponya offered a daring plan: with the few weapons they had among them, Ponya would lead a small group to attack the Vanduul and draw their attention away. In whatever time they could buy, the rest of the workers could continue their escape and hopefully make it to the nearest evac zone.

Ponya bid goodbye to her husband and to her daughter, Cedra, who was only seven at the time. It was the last time Cedra ever saw her mother but, thanks to the bravery of Ponya and those that followed her, the remaining families from their factory were able to make it to the nearest evacuation zone, joining the more than a million civilians who were escorted to safety by Squadron 214.

DAWN OF A NEW DREAM

Like many evacuees, Cedra Selea found herself resettled on Aremis in the neighboring Vega system. There, she and her father struggled to rebuild a new life for themselves out of the hardships they had endured. Due to the trauma of her escape, Cedra never forgot the devastation wrought by the Vanduul though found community among the other displaced refugees who shared her predicament.

When she was 19, Cedra realized the opportunity before her: Many of the refugees around her had been craftsmen in the factories back on Virgil and had years of experience making defensive equipment, yet local employers in Vega were reticent to capitalize on that expertise. Exhibiting some of the same natural leadership her mother had once shown, Cedra convinced a small cohort of these refugees to pool their resources and form a new company to meet the rapidly growing demand for personal armor. Together, they could work to achieve their desire to help Humanity be better prepared to face the Vanduul. They decided to name the company in tribute to their fallen home and so, in 2749, Virgil Limited was born.

RIGHT PLACE, RIGHT TIME

With demand vastly outpacing supply, Virgil's initial stock of armor quickly sold out. Even as the personal armor marketplace grew crowded by others seeking to exploit the same opportunity, the military-industrial influence on Virgil's designs helped the company stand out. Before long, the company expanded their production facility, hiring more skilled refugees from Virgil.

With Vega II acting as the new de-facto border to Vanduul-controlled space, the UEEN presence on the planet was significant and, eventually, the Empire caught wind of Virgil Limited's work. Within a few years of their start, Cedra secured their first military contract – a large retainer that allowed them to continue scaling ever faster.

They reliably fulfilled their contracts with the UEE and, eventually, were commissioned to develop the standard body armor for the Advocacy – a feat that spread their name around all of UEE space. The armor they developed became the foundation of the now widely popular TrueDef armor line, which is known for providing an ideal balance of protection and mobility.

Virgil also gained renown for their innovations in the production of power-armor commonly known as "Titan Suits"; popular but expensive equipment intended for heavy commercial and industrial applications, such as mining and cargo management.

RALLYING CRY

Bolstered by military contracts and success in both the commercial and industrial sectors, Virgil enjoyed many years of steady success. Of course, everything changed again on October 2945 when the Vanduul invaded the Vega System, unleashing a planet-wide attack on Aremis. There were many casualties on the planet, including all of Virgil's senior leadership, who died when the company headquarters were destroyed as part of the devastation of New Corvo.

It was an unfortunate echo of the events that led to Virgil's founding. Suffering from the loss of leadership and infrastructure, many thought the company might fail, but when the dust settled the company treated the disaster as a rallying cry, reaffirming their original mission and purpose: to help Humanity survive the Vanduul threat. Under the leadership of the new CEO, Vash Derrin, a former mid-level manager and a descendant of the company's early employees, Virgil has evolved once more, transitioning to a decentralized mode of operation, and selling off lesser patents and investments to finance the rebuilding of its destroyed infrastructure. While this period of reconstruction is still ongoing, it's clear that Virgil Limited isn't done fighting yet.