Empire Scale: Architectural Paradigms for Autonomous Inter-Room Colonization

The transition from a high-functioning Room Control Level (RCL) 8 colony to a multi-room empire represents the most significant architectural hurdle in Screeps AI development. At RCL 8, a system has optimized its local energy throughput, utilizing three spawns, sixty extensions, and a complex network of links, storage, and terminals to maintain peak efficiency.[1] However, the progression of the Global Control Level (GCL)—which governs the maximum number of claimable rooms and the total CPU limit—requires the continuous export of surplus energy into new controllers.[1] This expansion must be managed not as a series of manual interventions, but as a fully automated lifecycle driven by strategic directives, mathematical surplus verification, and clearly defined maturation thresholds.

Architectural Foundations of the Expansion Engine

The architecture of a successful colonization system is predicated on the underlying game loop, where player scripts are executed within a Node.js VM context once per tick.[2] Because each tick involves a sequence of script calculation followed by command processing, the AI must be designed for maximum asynchronous efficiency, utilizing the Memory object for persistent state between ticks while minimizing redundant calculations through caching.[2]

The Directive-Overlord Hierarchy

The senior architect must implement a hierarchical control model to manage the complexities of colonization. This model typically utilizes a ColonizeDirective as the primary state container. In high-level frameworks like Overmind, a Directive serves as a wrapper for a game flag, providing a geographic and contextual anchor for logic.[4] These directives are instantiated by a top-level Overseer or Global Manager that scans the environment for expansion opportunities or responds to user-placed flags.[4]

Attached to these directives are Overlords (or Managers), which are specialized modules responsible for the spawning and tasking of creeps to fulfill the directive's goal.[4] For expansion, this involves a "Bootstrapping Overlord" that coordinates the parent room's resources to support the child room's development until it reaches self-sufficiency.[4]

Global Control Level (GCL) and CPU Scaling

A critical architectural constraint is the Global Control Level. To control multiple rooms, the GCL must be sufficiently high (e.g., GCL 3 is required for 3 rooms).[1] Furthermore, on official servers, GCL determines the CPU limit, adding 10 CPU per level up to a maximum of 300.[1] As an empire expands, the CPU efficiency of each room becomes paramount. The senior architect must design colonization logic that minimizes expensive operations like find calls and pathfinding by caching WorldState data and reusing paths.[3]

The Lifecycle of the ColonizeDirective

The ColonizeDirective follows a linear, state-driven lifecycle that transitions the target room from a neutral state to a functional RCL 3 outpost. This lifecycle is designed to minimize the risk of "Colony Crash" (a total loss of population) by maintaining a continuous flow of labor and resources from the parent colony.[4]

Phase I: Scouting and Environmental Assessment

The lifecycle begins with the placement of a flag, which triggers the instantiation of the ColonizeDirective. If the target room is not currently visible, the directive enters a SCOUTING state. The parent colony spawns a fast, low-cost Scout creep (typically [MOVE]) to gain vision.[6]

During this phase, the directive performs a comprehensive analysis of the room's terrain and resource potential. The "Architect" module uses algorithms like Distance Transform and Flood Fill to determine if the room can accommodate a standard "Bunker" or "Stamp" layout.[8] High-priority targets are rooms with two energy sources, as these provide a maximum gross income of 6,000 energy per 300 ticks (20 energy per tick).[9] The directive also assesses the proximity to minerals; a room that provides access to a missing mineral in the empire's production chain (such as Hydrogen or Oxygen for XGH2O compounds) is highly valued.[8]

Phase II: The Legal Claim (The Lawyer Phase)

Once vision is secured and the room is validated, the directive transitions to the CLAIMING state. The parent colony spawns a "Lawyer" or Claimer creep. This creep is specifically designed to navigate to the target room and execute the creep.claimController() method.[12]

Body Part	Cost	Effect on Claiming
CLAIM	600	Claims, reserves, or attacks a controller.[7]
MOVE	50	Reduces fatigue by 2 per tick.[7]

The claimer must reach the controller before its 1,500-tick lifespan expires.[6] Successful claiming immediately changes the room's status to "Owned," allowing the construction of

OwnedStructures like Spawns and Extensions.[1] Crucially, claiming doubles the energy capacity of local sources from 1,500 to 3,000.[11]

Phase III: The Pioneer Influx (The Surrogate Phase)

With ownership established, the room is at RCL 1 but lacks the capacity to build a Spawn locally, as a Spawn costs 15,000 energy and the initial energy capacity of the room is only 300.[1] The directive enters the BOOTSTRAPPING state, where the parent colony at RCL 8 spawns "Pioneer" creeps.

Pioneers are massive, 50-part workers that carry their own starting energy and possess high WORK counts to accelerate the building of the first spawn.[6] These creeps act as a bridge, performing harvesting, building, and upgrading simultaneously. The parent colony must maintain a "Pioneer Queue," ensuring that as one Pioneer dies or nears the end of its life, another is dispatched to take its place. This prevents a gap in labor that could lead to the controller's downgrade timer triggering.[14]

Phase IV: Infrastructure Milestones (RCL 1 to RCL 3)

The goal of the ColonizeDirective is to reach RCL 3, which unlocks the first Tower.[1] The sequence of construction is critical for survival:

1. **Spawn #1**: The highest priority. Once built, the room can begin spawning its own small workers.[1]
2. **Extensions (1-10)**: These increase the local energy capacity from 300 to 800, allowing for more effective local creeps.[1]
3. **Tower**: The security milestone. A functional tower defends against early invaders and can repair decaying structures.[14]

During this phase, the room is in a "Surrogate" relationship with the parent. If local mining is insufficient, the parent may send "Transporters" to deliver bulk energy until the local economy stabilizes.[13]

Death Spiral Prevention: The Mathematical Calculus of Surplus

A death spiral occurs when the maintenance and expansion costs of the empire exceed its total production, leading to a cascade of colony crashes.[19] For a Senior Architect, preventing this requires a formal mathematical verification before any expansion attempt is initiated.

The Net Energy Surplus Formula

The parent colony must calculate its Net Energy Surplus ($E_{surplus}$) to determine if it can support a child without compromising its own stability. This surplus is defined by the total income from local and remote sources minus the mandatory expenditures.

$$E_{surplus} = \sum(I_{local} + I_{remote}) - (X_{spawn} + X_{repair} + X_{upgrade\_min})$$

Where:

- I is the energy income per tick (e.g., 10 per source).[9]
- X_{spawn} is the energy cost of replacing existing creeps divided by their lifespan (1,500 ticks).[6]
- X_{repair} is the energy required to maintain roads, containers, and ramparts.[7]
- $X_{upgrade\_min}$ is the 15 energy per tick required to prevent an RCL 8 controller from downgrading.[1]

The Expansion Investment Threshold

Expansion is only authorized if $E_{surplus} > C_{expansion\_rate}$, where $C_{expansion\_rate}$ is the energy per tick required to sustain the expansion effort. The architect should also use a Storage threshold as a physical buffer. Expansion should be paused if the parent's storage drops below a safety level (e.g., 100,000 energy).[10]

The cost of a single pioneer creep ($C_{pioneer}$) must be amortized over its travel time (T_{travel}) and work time (T_{work}).

$$C_{tick\_cost} = \frac{Cost_{body} + (WORK_{parts} \times Energy_{build\_per\_tick} \times T_{work})}{1500}$$

If the parent cannot afford this C_{tick_cost} without dipping into its minimum reserves, the colonization request is queued until energy levels recover.

CPU as a Limiting Factor

In the context of death spirals, CPU is often more dangerous than energy. An expansion attempt that consumes too much CPU for pathfinding or scouting can cause the entire AI to exceed its bucket, leading to missed ticks and a failure to issue move or harvest commands.[2] The architect must monitor the Game.cpu.getUsed() and Game.cpu.bucket metrics. If the bucket is low, the expansion directive should enter a HIBERNATION state, reducing its scouting and spawning frequency until the CPU buffer is replenished.[14]

Maturation: Transitioning from Dependent to Autonomous

The transition of a new colony from a "Dependent Child" to an "Autonomous Adult" is a precise architectural handover. This handover occurs at specific infrastructure and security milestones that ensure the room can survive and grow without external help.

The Milestone of Defensive Autonomy (RCL 3)

A room is no longer a liability when it achieves defensive autonomy. This is reached at **RCL 3 with a completed and energized Tower**.[1]

- **Invader Dynamics**: Up to RCL 3, NPC invaders are "Light" and can be handled by a single tower.[17]
- **Repair Autonomy**: The tower can repair containers and roads, preventing the loss of mining infrastructure to decay.[7]
- **Security handover**: At this point, the parent colony can withdraw its military "Guards" from the room, as the tower provides 150–600 damage per tick to any point in the room.[14]

The Milestone of Economic Autonomy (RCL 4)

True autonomy—the point where the colony is considered an "Adult"—is reached at **RCL 4 with a completed Storage**.[1]

- **The Storage Buffer**: Storage allows the room to hold up to 1,000,000 energy.[1] This decouples energy consumption from instantaneous mining, allowing the room to survive temporary source exhaustion or combat interruptions.[22]
- **Heavy Invaders**: At RCL 4, invaders transition to "Heavy" squads which may include boosted healers.[17] Economic autonomy is defined by the room's ability to spawn its own "Defender" creeps (A12M6 or similar) using the energy in its storage and extensions.[24]
- **Spawning Autonomy**: With 20 extensions, the room has a capacity of 1,300 energy (300 from spawn + 20 $\times$ 50 from extensions).[1] This allows the room to spawn specialized WORK-heavy harvesters (5 WORK parts) that can fully exhaust a source.[10]

The precise handover point

The Senior Architect marks a colony as "Autonomous" when:

1. The RCL is $\geq$ 4.
2. A STRUCTURE_STORAGE exists with at least 10,000 energy.
3. The room possesses at least one STRUCTURE_SPAWN.
4. The local harvester and hauler population can be sustained by the local spawn's energy capacity.

At this point, the ColonizeDirective is deleted, and the room is transitioned to a standard Colony process in the AI's global registry.[4]

Optimal Pioneer Design for Parent Room Spawning

When spawning pioneers from an RCL 8 room, the architect can utilize the full 50-part limit and 12,900 energy capacity.[1] The design must maximize the energy-per-tick (EPT) the pioneer can apply to construction sites.

Fatigue and Movement Mechanics

Movement speed is governed by the ratio of MOVE parts to other body parts. Fatigue is generated by the weight of non-move parts and terrain factors.[7]

$$Fatigue_{per\_move} = 2 \times (Weight \times TerrainFactor - MOVE_{parts})$$

Where:

- $Weight$ is the number of body parts (excluding MOVE and empty CARRY).[7]
- $TerrainFactor$ is 0.5 for roads, 1 for plains, and 5 for swamps.[7]

Pioneer Body Patterns

To ensure a pioneer can reach a new room and begin working efficiently, the body must be optimized for speed and capacity.

Pioneer Type	Body Composition	Cost	Capabilities
All-Terrain	25 WORK, 25 MOVE	3,750	125 build pts/tick; moves 1/tick on plains.[6]
Heavy Builder	30 WORK, 10 CARRY, 10 MOVE	4,000	150 build pts/tick; requires roads for 1/tick speed.[7]
Remote Settler	15 WORK, 10 CARRY, 25 MOVE	2,750	75 build pts/tick; high capacity for swamp navigation.[6]

A pioneer with 25 WORK parts will consume 25 energy per tick to build, or 50 energy per tick to upgrade at RCL 1–7 (at RCL 8, upgraders are capped at 15/tick).[1] The architect must ensure the child room's sources can provide at least as much energy as the pioneers consume, or the parent must provide a continuous logistics train.

Advanced Infrastructure: Scaling to the Terminal Network

Once a new colony reaches RCL 6, the colonization architecture shifts from survival to empire integration. This is marked by the construction of the Extractor, Labs, and most importantly, the Terminal.[1]

The Empire Logistics Network

The Terminal allows for the transfer of resources between rooms for a cost in energy.[1] This is the "Endgame" of colonization architecture.

- **Inter-room energy balancing**: Terminals can automatically send energy from high-surplus rooms to struggling new colonies, effectively bypassing the need for physical pioneer travel.[11]
- **Mineral Consolidation**: New colonies can send their local minerals to a central "Hub" room for processing into advanced boosts like XGH2O.[8]
- **Market Dealing**: Colonies can buy or sell resources on the global market to fund their growth.[11]

Leaky Bucket Throughput

The architect must account for the terminal throughput limits implemented in newer server versions. Inbound terminal transactions are limited to a "leaky bucket" of 50 units per tick, accumulating up to 300,000.[28] This prevents a single room from being instantly flooded with millions of units of energy, reinforcing the need for each colony to maintain its own strong harvesting infrastructure even after autonomy.

Strategic Room Selection and Territory Intelligence

The final component of a colonization system is the automated selection of targets. A Senior Architect does not wait for a flag; the AI should use the Observer structure (unlocked at RCL 8) to scan the world map.[1]

The Selection Algorithm

Rooms are scored based on several metrics to ensure the highest return on energy and CPU investment:

1. **Source Proximity**: Sources should be close to the controller or a central bunker site to

minimize hauler travel time and CPU cost.[9]
2. **Terrain Accessibility**: Rooms with large, contiguous areas of plain terrain are preferred for bunker layouts, as they require fewer ramparts and have lower pathfinding costs.[8]
3. **Strategic Isolation**: Rooms in "Novice Areas" or "Respawn Areas" are temporarily isolated by walls, providing a safe period for expansion.[30]
4. **Mineral Synergy**: The AI tracks the distribution of minerals across the empire. If the empire lacks Zynthium, the colonization engine will prioritize a target room with a Zynthium deposit.[8]

Handling Controller Blockage and Hostility

The colonization engine must be resilient to interference. If a controller is signed with a hostile message or is being reserved by another player, the ColonizeDirective must decide whether to escalate (spawn a Dismantler or Attacker) or concede and select a new target.[12] The architect should implement a "Cool-down" period for rooms where colonization failed, preventing the infinite waste of creeps on a contested or unviable room.[13]

Conclusion: The Architecture of Infinite Growth

The colonization of new rooms in Screeps is the ultimate test of an AI's architectural robustness. By codifying the lifecycle of a ColonizeDirective, implementing strict mathematical thresholds for surplus, and defining clear maturation milestones at RCL 3 and 4, the Senior Architect creates a system that can scale indefinitely. This engine transforms the empire from a collection of isolated rooms into a singular, distributed intelligence, capable of funneling the vast energy of RCL 8 colonies into the next generation of expansion. Through the careful management of energy, CPU, and defensive risk, the automated empire can thrive in the persistent, competitive world of Screeps, moving relentlessly toward the peak of Global Control Level dominance.

Works cited

1. Control | Screeps Documentation, accessed February 19, 2026, https://docs.screeps.com/control.html
2. Server-side architecture overview | Screeps Documentation, accessed February 19, 2026, https://docs.screeps.com/architecture.html
3. Screeps #12: Strategic Directives | Field Journal, accessed February 19, 2026, https://jonwinsley.com/notes/screeps-strategic-directives
4. Screeps #1: Overlord overload - Ben Bartlett, accessed February 19, 2026, https://bencbartlett.com/blog/screeps-1-overlord-overload/
5. Great Filters - Screeps Wiki, accessed February 19, 2026, https://wiki.screepspl.us/Great_Filters/
6. Creeps - Screeps Documentation, accessed February 19, 2026, https://docs.screeps.com/creeps.html
7. Creep | Screeps Wiki | Fandom, accessed February 19, 2026,

https://screeps.fandom.com/wiki/Creep
8. Getting Started - Screeps Wiki, accessed February 19, 2026, https://wiki.screepspl.us/Getting_Started
9. Screeps #1: The Game Plan | Field Journal, accessed February 19, 2026, https://jonwinsley.com/notes/screeps-game-plan
10. Is there a guide out there for the ideal ratios of your creeps? : r/screeps - Reddit, accessed February 19, 2026, https://www.reddit.com/r/screeps/comments/864qy3/is_there_a_guide_out_there_for_the_ideal_ratios/
11. Energy - Screeps Wiki, accessed February 19, 2026, https://wiki.screepspl.us/Energy/
12. StructureController - Screeps Wiki, accessed February 19, 2026, https://wiki.screepspl.us/StructureController/
13. Screeps #19: Expanding Operations - Field Journal, accessed February 19, 2026, https://jonwinsley.com/notes/screeps-expanding-operations
14. Getting Started - Screeps Wiki, accessed February 19, 2026, https://wiki.screepspl.us/Getting_Started/
15. Room Control Level - Screeps Wiki, accessed February 19, 2026, https://wiki.screepspl.us/Room_Control_Level/
16. How many body parts can I build? - screeps - Reddit, accessed February 19, 2026, https://www.reddit.com/r/screeps/comments/7nlm3x/how_many_body_parts_can_i_build/
17. NPC Invaders | Screeps Documentation, accessed February 19, 2026, https://docs.screeps.com/invaders.html
18. Faster Controller Levelling : r/screeps - Reddit, accessed February 19, 2026, https://www.reddit.com/r/screeps/comments/5jia18/faster_controller_levelling/
19. Applied Energy Death spiral, transmission charges, and prosumers in the electricity market, accessed February 19, 2026, https://par.nsf.gov/servlets/purl/10409979
20. Electric Utilities' 'Death Spiral': Hyperbole or Reality?, accessed February 19, 2026, https://energia.pr.gov/wp-content/uploads/sites/7/2016/07/Attachment-RH-1-Costello-Hemphill-Death-Spiral-2014_Final-Pub.pdf
21. Overmind - Ben Bartlett, accessed February 19, 2026, https://bencbartlett.com/overmind-docs/
22. Storage change | Screeps Forum, accessed February 19, 2026, https://screeps.com/forum/topic/178/storage-change
23. what do people use containers and storage for? | Screeps Forum, accessed February 19, 2026, https://screeps.com/forum/topic/640/what-do-people-use-containers-and-storage-for
24. RCL 4 Boosted Invaders | Screeps Forum, accessed February 19, 2026, https://screeps.com/forum/topic/2193/rcl-4-boosted-invaders
25. How MOVE part work？ | Screeps Forum, accessed February 19, 2026, https://screeps.com/forum/topic/1314/how-move-part-work
26. Gaps in the Documentation : r/screeps - Reddit, accessed February 19, 2026,

https://www.reddit.com/r/screeps/comments/2pqhsz/gaps_in_the_documentation/
27. Workflow tips and prioritization for new players? | Screeps Forum, accessed February 19, 2026, https://screeps.com/forum/topic/2556/workflow-tips-and-prioritization-for-new-players
28. Discussion: long-range logistics revamp | Screeps Forum, accessed February 19, 2026, https://screeps.com/forum/topic/2975/discussion-long-range-logistics-revamp
29. Screeps #15: Room Planning | Field Journal, accessed February 19, 2026, https://www.jonwinsley.com/notes/screeps-room-planning
30. Start Areas | Screeps Documentation, accessed February 19, 2026, https://docs.screeps.com/start-areas.html
31. Resources | Screeps Documentation, accessed February 19, 2026, https://docs.screeps.com/resources.html