Engineering the High-Tier Economy: Automated Link and Terminal Management in Screeps Architecture

The transition from a developing colony to a high-tier economic powerhouse in Screeps is defined by the shift from labor-intensive logistics to infrastructure-driven resource distribution. At Room Controller Level (RCL) 1 through 4, the economy relies on "Type Zero" harvesting, where mobile units (creeps) physically transport energy from sources to storage containers. However, upon reaching RCL 5 and 6, the introduction of the StructureLink and StructureTerminal allows for a fundamental decoupling of resource throughput from creep movement. This report provides an exhaustive engineering analysis of automated link and terminal management, establishing best practices for link balancing logic, market API automation, and empire-wide resource synchronization.

The Physicality of Instantaneous Transfer: StructureLink Mechanics

The StructureLink represents the first tier of high-efficiency logistics, enabling the instantaneous transfer of energy across room terrain without the latency associated with pathfinding. While links eliminate the need for courier creeps, they are governed by rigid physical and mathematical constraints that must be accounted for in any automated management script.

Spatial Constraints and Throughput Mathematics

A link pair's efficiency is inversely proportional to the linear distance between the sender and the receiver. Every transfer triggers a cooldown period calculated in game ticks, equivalent to the range between the two structures. This cooldown dictates the maximum theoretical bandwidth of a link network. For example, a link pair separated by 50 tiles has a maximum throughput of 800 energy units every 50 ticks, or 16 energy units per tick. Given that a standard energy source produces 10 units per tick (or 3000 units every 300 ticks), a single link can comfortably handle the output of one or even two sources provided they are within reasonable proximity to the central hub.

Furthermore, links incur a mandatory energy loss, often referred to as a "transmission tax." This loss is calculated as:

where A represents the amount of energy transferred. Despite this 3% loss, links remain significantly more efficient than creep-based hauling when accounting for the energy cost of spawning couriers, the decay of road infrastructure, and the CPU overhead of pathfinding calculations.

Link Role Categorization and Identification

A top-tier automated system does not treat all links as generic containers. Instead, it employs a role-based identification system, typically executed during the room's initialization or upon an

RCL increase.

Link Role	Description	Geographic Placement	Priority
Source Link	Acts as a primary producer; collects energy from local harvesters.	Adjacent to Energy Sources.	Low (Sender)
Hub Link	The central node of the network; typically adjacent to the room’s Storage.	Within Range 1 of StructureStorage.	Medium (Buffer)
Controller Link	Supplies energy to static upgrader creeps.	Within Range 3 of the Room Controller.	High (Receiver)
Tower Link	Ensures towers remain functional for defense without dedicated fillers.	Adjacent to defensive clusters.	Critical (Receiver)
Extension Link	Optimizes the filling of distant extension clusters during spawn bursts.	Central to an extension farm.	Medium (Receiver)
Remote Link	Receives energy from long-distance haulers at the room’s edge.	Near room exits/entrances.	Low (Sender)

The identification process leverages Room.find(FIND_MY_STRUCTURES) filtered by STRUCTURE_LINK, followed by proximity checks against key room objects like the Storage or Controller. Once identified, these links are stored in the room's global memory to avoid expensive re-calculation in subsequent ticks.

Algorithmic Patterns for Link Balancing Logic

The goal of automated link management is to ensure that energy is always available at high-priority "Sinks" (e.g., the Controller Link) while preventing "Sources" (e.g., Source Links) from reaching capacity and halting production.

The Hub-and-Spoke Priority System

The most robust architectural pattern for link management is the Hub-and-Spoke model. In this configuration, the Hub Link (adjacent to Storage) acts as the primary distributor. The logic flow is segmented into two distinct phases: Collection and Distribution.

During the Collection phase, all Source and Remote links evaluate their internal energy levels. When a sender reaches a specific threshold (typically 700-800 energy), it initiates a transfer to the Hub Link. This "push" logic ensures that harvesting sites remain clear for incoming energy.

In the Distribution phase, the Hub Link scans for available Sinks based on a predefined priority queue.

1. **Tower Links:** If a Tower Link’s energy falls below 400, the Hub Link prioritizes a transfer to ensure defensive readiness.
2. *Controller Link:* If the Controller Link falls below a threshold (e.g., 200 energy), the Hub

Link transfers energy to sustain upgrading.
3. *Extension Link:* During spawning cycles, if the Hub Link has surplus energy and the Extension Link is empty, a transfer is executed to assist filler creeps.

The Balanced Network Approach

In complex base layouts where links may serve multiple purposes (e.g., a link that both receives from a remote source and supplies a nearby tower), a "Balancing" algorithm is preferred. This method calculates the average energy across all links in the room and attempts to equalize them. While mathematically elegant, it is less energy-efficient than priority-based systems due to the 3% transfer tax; moving energy multiple times to achieve "balance" results in higher cumulative loss. Therefore, balancing should only be applied to multi-use nodes while dedicated source and sink links remain within a priority hierarchy.

Terminal Logistics: Global Trade and Empire-Wide Synchronization

Upon reaching RCL 6, the introduction of the StructureTerminal expands the economy from a local room-scale to a global world-scale. Terminals allow for the instantaneous transfer of any resource between any two rooms on the map, provided both possess a terminal.

The Leaky Bucket Throughput Model

A critical recent development in terminal mechanics is the "Leaky Bucket" throughput limit. Incoming transactions are restricted to a specific volume per tick—preliminary values suggest 50 units per tick—which can accumulate up to a burst capacity of 300,000 units. Automated logistics systems must track this "available throughput" to prevent deal failures during high-intensity operations like GCL farming or "room burst upgrading," where multiple rooms might attempt to send energy to a single destination simultaneously.

Transaction Cost Calculus and Distance Optimization

Transferring resources via terminal is not free; it consumes energy based on the amount of resources and the linear distance (d) between the origin and destination. The energy cost is defined by the following non-linear relationship:
This formula implies that as distance increases, the energy cost approaches the total amount of resources being sent. At d = 30, the cost is approximately 63% of the amount; at d = 60, it reaches 86%. Consequently, an automated terminal governor must prioritize local transfers over distant ones. Empire-wide resource balancing logic should always seek to satisfy a room's deficit using the closest available surplus to minimize the energy overhead of the trade network.

Market API Automation: Growing the Credit Balance

A top-tier economy utilizes the Screeps Market to convert surplus minerals and commodities into Credits, which can then be used to purchase lacking resources or high-tier boosts.

Price History and Trend Analysis

Automation scripts should leverage the Game.market.getHistory(resourceType) method to retrieve 14-day price data. This data allows for the implementation of advanced trading indicators such as Simple Moving Averages (SMA) or Exponential Moving Averages (EMA) to identify "fair" market value.

Market Strategy	Description	Implementation Step
Mean Reversion	Buying assets when they are significantly below their historical average price.	Monitor getHistory for prices < SMA(14) * 0.9.
Momentum Trading	Capitalizing on upward price trends for commodities like Power or Ghodium.	Identify resources where current price > EMA(14).
NPC Arbitrage	Selling processed commodities to NPC terminals at highway crossroads.	Track NPC buy prices for "Bars" (e.g., Utrium Bar).

Order Management and Fee Optimization

Participating in the market requires a balance between "Deals" and "Orders". Using Game.market.deal() is immediate but requires the player to pay the energy cost and the terminal cooldown. Creating an order using Game.market.createOrder() avoids the energy cost but incurs a 5% credit fee and is subject to market competition.
A sophisticated Market Manager process should follow these best practices:

- **Order Throttling:** Limit the number of active orders to stay within the 300-order limit per shard.
- **Price Adjustment:** Periodically call Game.market.changeOrderPrice() to remain competitive, but be wary of the 5% tax on price increases for buy orders.
- **Volume Extension:** Use Game.mark[span_48](start_span)[span_48](end_span)et.extendOrder() to replenish successful sell orders rather than creating new ones, which maintains the order's presence without resetting its aging.

High-Tier Economic Structures: Factories and Commodities

At RCL 7 and 8, the economy moves beyond raw minerals into the production of commodities. Factories process base minerals and energy into highly compressed goods (e.g., Batteries, Bars, and complex commodities like Microchips).

The Value-Added Chain

Commodities serve two primary purposes: storage compression and credit generation. Processing 500 units of energy into 50 Batteries reduces the storage footprint significantly while providing a portable form of energy for terminal transfers. More importantly, high-level commodities are the only resources NPC traders consistently purchase at high prices, making them the primary engine for credit growth in the endgame.

Product	Ingredients	Factory Level	NPC Interest
Energy Battery	600 Energy	Any	Low (Compression only)
Utrium Bar	500 Utrium + 200 Energy	Any	High (Standard sink)
Cell	500 Biomass + 200 Energy	Any	High (Quadrant specific)
Tier 5 Commodity	Complex Reagents	Level 5	Critical (Max Credits)

Automated factory management requires tight integration with terminal logistics. The "Factory Overlord" must request reagents from the empire-wide network and ensure that finished products are either stored for future use or sent to highway rooms for sale to NPC terminals.

Software Architecture: The Economic Kernel

To manage links, terminals, and the market efficiently across multiple rooms, a process-based Operating System (OS) architecture is essential. This approach prevents the "starvation" of critical logic and ensures that CPU usage remains within the limits provided by GCL levels.

Process Decomposition and Priority Scheduling

The economic system should be broken down into discrete processes, each managed by a scheduler that allocates CPU based on urgency.

1. **Link Process (Priority: Critical):** Executed every tick. Handles local energy transfers to towers and controllers to ensure room survival.
2. **Logistics Process (Priority: High):** Executed every 5-10 ticks. Manages the movement of energy between Storage and Terminal within a room.
3. **Terminal Process (Priority: Medium):** Executed every 20 ticks. Synchronizes resource levels across the empire and fulfills internal requests.
4. **Market Process (Priority: Low):** Executed every 100 ticks. Scans getAllOrders, updates prices, and executes profitable trades.

By decoupling these tasks, the bot can skip the Market Process if the CPU bucket is low without risking the room's defense (managed by the Link Process).

The Terminal Overlord Pattern

In a multi-room colony, a central "Terminal Overlord" process maintains a global view of all assets. It calculates an "Empire Average" for each resource and identifies rooms with significant deviations. If Room A has 800,000 energy (surplus) and Room B has 20,000 (deficit), the Overlord calculates the transaction cost and issues a send command. This centralized management prevents "ping-ponging" where two rooms repeatedly send resources back and forth due to conflicting local logic.

Advanced Optimization: Power Creeps and Storage Expansion

The ultimate evolution of automated management involves the deployment of Power Creeps

(PCs). These hero units possess unique abilities that drastically enhance the efficiency of links and terminals.

- **OPERATE_LINK:** Decreases link cooldown and increases energy transfer efficiency.
- **OPERATE_TERMINAL:** Decreases the energy cost of terminal transfers and reduces the cooldown.
- **OPERATE_STORAGE:** Temporarily increases the storage capacity by up to 7 million units, allowing a room to act as a massive economic buffer during market fluctuations.
- **REGEN_SOURCE:** Increases the energy output of a source, necessitating higher-throughput link balancing logic to prevent overflow.

An automated system must coordinate PC movement to ensure these buffs are applied precisely when needed—for instance, activating OPERATE_TERMINAL just before a large-scale empire-wide redistribution of energy.

Risk Mitigation and Resource Locking

High-tier automation faces risks such as "Terminal Congestion" and "Credit Exhaustion".

Terminal Congestion and Deadlocks

A terminal can become "deadlocked" if it is filled with minerals but lacks the energy required to pay the transfer fee to send them away. To prevent this, best practices dictate the use of a "Minimum Energy Reserve." The terminal should always lock 20,000 to 50,000 units of energy that cannot be consumed by factories or the market, ensuring the logistics network remains operational.

Market Volatility and Safety Buffers

Automated trading algorithms must implement "Stop-Loss" and "Slippage" protections. Before executing a Game.market.deal(), the script should compare the current price against the 14-day history. If the price deviates by more than 2 standard deviations (a "Z-Score" check), the trade should be aborted as it likely represents a momentary market anomaly or a predatory order.

Conclusion: Implementing the Top-Tier Economy

Achieving a top-tier economy in Screeps requires the seamless integration of spatial planning, mathematical optimization, and algorithmic trading. By transitioning to a role-based link priority system, players eliminate the CPU waste of mobile haulers and ensure that room-critical structures are always powered. Moving to a terminal-centric empire allows for the efficient concentration of resources, enabling the production of high-value commodities and the steady growth of a credit balance. The success of this implementation is measured not by the amount of energy harvested, but by the "Credit Velocity" and the stability of the colony under stress. As the bot evolves, the focus shifts from managing creeps to managing flows, turning the empire into a self-sustaining autonomous machine capable of dominating the global market.

Works cited

1. Control | Screeps Documentation, https://docs.screeps.com/control.html 2. Great Filters -

Screeps Wiki, https://wiki.screepspl.us/Great_Filters/ 3. Link | Screeps Wiki - Fandom, https://screeps.fandom.com/wiki/Link 4. StructureLink - Screeps Wiki, https://wiki.screepspl.us/StructureLink/ 5. Balance of Links | Screeps Forum, https://screeps.com/forum/topic/1744/balance-of-links 6. Introduce an upper limit for terminal transfers | Screeps Forum, https://screeps.com/forum/topic/935/introduce-an-upper-limit-for-terminal-transfers 7. Operating System - Screeps Wiki, https://wiki.screepspl.us/Operating_System/ 8. Concept Question: How to effectively use a Link and distribute Energy in the room : r/screeps, https://www.reddit.com/r/screeps/comments/4sy6f6/concept_question_how_to_effectively_use_a_link/ 9. Market System | Screeps Documentation, https://docs.screeps.com/market.html 10. Discussion: long-range logistics revamp | Screeps Forum, https://screeps.com/forum/topic/2975/discussion-long-range-logistics-revamp 11. Power | Screeps Documentation, https://docs.screeps.com/power.html 12. Market - Screeps Wiki, https://wiki.screepspl.us/Market/ 13. Intermediate-level tips - Screeps Wiki, https://wiki.screepspl.us/Intermediate-level_tips/ 14. How do you manage chemistry? | Screeps Forum, https://screeps.com/forum/topic/1895/how-do-you-manage-chemistry 15. Energy - Screeps Wiki, https://wiki.screepspl.us/Energy/ 16. Trading Algorithms: A Complete Step-By-Step Guide | Intrinio, https://intrinio.com/blog/how-to-create-a-trading-algorithm-essential-steps-for-success 17. Sma — Indikator dan Strategi - TradingView, https://id.tradingview.com/scripts/sma/ 18. Strona 7 | Wykładnicza średnia krocząca (EMA) — Wskaźniki i Strategie - TradingView, https://pl.tradingview.com/scripts/ema/page-7/ 19. Resources | Screeps Documentation, https://docs.screeps.com/resources.html 20. Draft: factories and commodities (new crafting/trading mechanic) | Screeps Forum, https://screeps.com/forum/topic/2557/draft-factories-and-commodities-new-crafting-trading-mechanic 21. Screeps #22: For Science | Field Journal, https://jonwinsley.com/notes/screeps-for-science 22. Screeps #1: Overlord overload - Ben Bartlett, https://bencbartlett.com/blog/screeps-1-overlord-overload/ 23. Screeps #26: World - Discrete Objectives | Field Journal, https://jonwinsley.com/notes/screeps-discrete-objectives 24. Automated Market Makers (AMMs): Math, Risks & Solidity Code | Speedrun Ethereum, https://speedrunethereum.com/guides/automated-market-makers-math 25. A detailed guide on Automated Market Maker AMM | by Rapid Innovation | Medium, https://medium.com/@rapidinnovation/a-detailed-guide-on-automated-market-maker-amm-a70cd80736ea