LEAN ETHEREUM: La Visión de una Década para el Protocolo de Consenso con Casper FFG y LMD-GHOST

De Gasper a Lean Consensus: Cómo Ethereum redefine su capa de consenso, datos y ejecución para lograr rendimiento extremo con seguridad post-cuántica inquebrantable

---

Fecha: Febrero 2026 **Versión:** 2.0 **Basado en:** "*lean Ethereum*" — Justin Drake, Ethereum Foundation Blog, 31 de julio de 2025

Tabla de Contenidos

---

1. Resumen (Abstract)

El 31 de julio de 2025 — un día después del décimo aniversario de Ethereum — **Justin Drake**, investigador de la Ethereum Foundation, presentó **Lean Ethereum**: una visión y misión personal para la próxima década del protocolo. La propuesta redefine fundamentalmente las tres capas de Ethereum L1 bajo un principio unificador de simplicidad, seguridad post-cuántica y rendimiento extremo.

Lean Ethereum se estructura en tres pilares de actualización: **Lean Consensus** (Beacon Chain 2.0, evolución de la propuesta Beam Chain), que transforma el protocolo Gasper actual — basado en Casper FFG y LMD-GHOST — para lograr finalidad en segundos; **Lean Data** (Blobs 2.0), que introduce blobs post-cuánticos con granularidad flexible; y **Lean Execution** (EVM 2.0), que propone un set de instrucciones mínimo y SNARK-friendly basado potencialmente en RISC-V. Transversalmente, **Lean Cryptography** unifica todo bajo criptografía basada en hashes, reemplazando las firmas BLS y los compromisos KZG por primitivas resistentes a computadoras cuánticas.

Los objetivos de rendimiento son ambiciosos: **1 gigagas/segundo en L1** (10,000 TPS) y **1 teragas/segundo en L2** (10 millones de TPS), manteniendo la descentralización y el 100% de uptime que Ethereum ha sostenido desde su génesis.

Este paper analiza en profundidad los fundamentos técnicos del protocolo de consenso actual (Casper FFG + LMD-GHOST = Gasper), explica cómo cada componente de Lean Ethereum lo transforma, y examina las implicaciones criptográficas, de seguridad y de escalabilidad de esta visión generacional.

Palabras clave: Lean Ethereum, Casper FFG, LMD-GHOST, Gasper, Beam Chain, post-quantum, hash-based cryptography, RISC-V, zkVM, Single Slot Finality, Lean Consensus, Lean Data, Lean Execution.

---

2. Introducción: ¿Qué es Lean Ethereum?

2.1 El Momento Fundacional

El 30 de julio de 2025, Ethereum cumplió 10 años. Al día siguiente, Justin Drake publicó en el blog oficial de la Ethereum Foundation un post titulado "*lean Ethereum*", articulando una visión comprehensiva para la próxima década. En sus propias palabras:

> "*We stand at the dawn of a new era. Millions of TPS. Quantum adversaries. How does Ethereum marry extreme performance with uncompromising security and decentralization?*"

La respuesta de Drake es **Lean Ethereum**: un marco de trabajo que reimagina las tres subcapas de Ethereum L1 desde primeros principios, con la criptografía basada en hashes como fundamento unificador.

2.2 De Beam Chain a Lean Ethereum

Lean Ethereum es la evolución natural de la **Beam Chain**, propuesta por el mismo Drake en noviembre de 2024 durante Devcon 7. Mientras que Beam Chain se enfocaba exclusivamente en un rediseño limpio de la capa de consenso, Lean Ethereum expande la visión para abarcar las tres capas:

```
Beam Chain (Nov 2024)          Lean Ethereum (Jul 2025)
─────────────────────          ────────────────────────

Solo consenso          →→→     Consenso + Datos + Ejecución
Rediseño del Beacon    →→→     Reimaginación completa de L1
Chain                          desde primeros principios
```

2.3 Los Dos Modos: Fort y Beast

Drake estructura su visión en dos imperativos complementarios:

```
┌──────────────────────────────────────────────────────────┐
│                      LEAN ETHEREUM                       │
├──────────────────────────────┬───────────────────────────┤
│        🛡 FORT MODE          │        ⚡ BEAST MODE      │
│        (Defensa)             │        (Ofensiva)         │
├──────────────────────────────┼───────────────────────────┤
│ Sobrevivir a los adversarios │ Rendimiento extremo       │
│ más hostiles:                │                           │
│ • Estados-nación             │ • 1 gigagas/s en L1 (10K TPS)  │
│ • Computadoras cuánticas     │ • 1 teragas/s en L2 (10M TPS)  │
│ • Ataques sostenidos         │ • Finalidad en segundos   │
│                              │ • zkVMs en tiempo real    │
│ Herramienta clave:           │ Herramienta clave:        │
│ Criptografía post-cuántica   │ SNARKs + escalado vertical│
│ basada en hashes             │ y horizontal              │
└──────────────────────────────┴───────────────────────────┘
```

2.4 Estructura de este Paper

Este documento se organiza en dos partes conceptuales:

Parte I (Secciones 3-6): Los cimientos — explica en profundidad el protocolo de consenso actual (GHOST, LMD-GHOST, Casper FFG, Gasper) que Lean Ethereum busca transformar. Comprender Gasper es requisito previo para entender por qué y cómo Lean Ethereum lo evoluciona.

Parte II (Secciones 7-17): La transformación — analiza cada pilar de Lean Ethereum, la criptografía post-cuántica que lo habilita, los ataques que resuelve, y la filosofía de diseño que lo sustenta.

PARTE I — LOS CIMIENTOS: EL PROTOCOLO DE CONSENSO ACTUAL

---

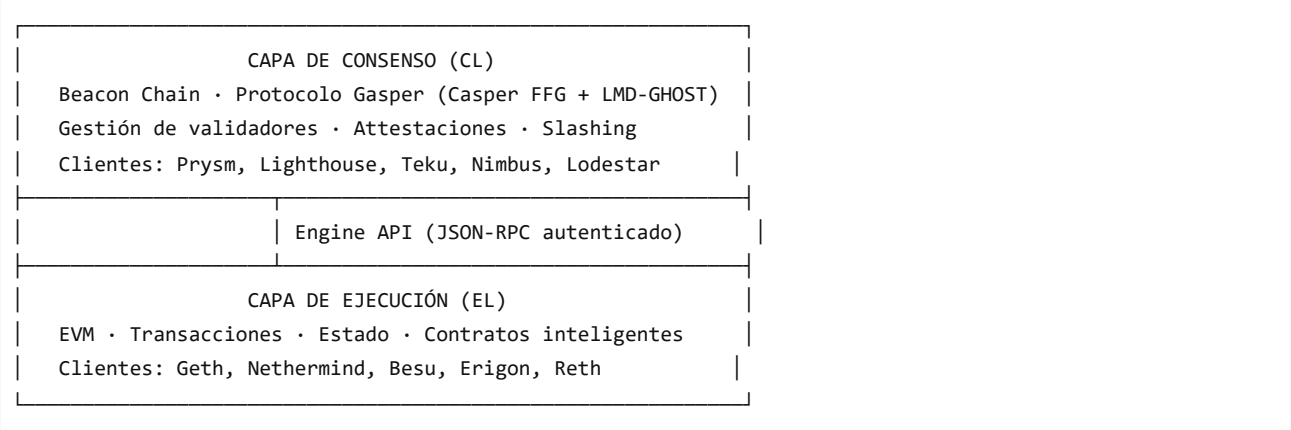
3. Contexto: De Proof-of-Work a Gasper

3.1 La Transición Histórica: The Merge

Ethereum nació en julio de 2015 usando Proof-of-Work (PoW) con el algoritmo Ethash. El **15 de septiembre de 2022**, en el bloque de ejecución 15,537,393 y el slot 4,700,013 de la Beacon Chain, Ethereum ejecutó **The Merge**: la transición a Proof-of-Stake (PoS) sin interrupción del servicio, reduciendo el consumo energético en un **99.95%**.

3.2 Arquitectura Post-Merge (Estado Actual)

Desde The Merge, Ethereum opera como dos capas acopladas:



3.3 Métricas del Sistema Actual (pre-Lean)

Métrica	Valor actual	Objetivo Lean
Validadores activos	~1,000,000+	Más participación (min. 1 ETH)
ETH stakeado	~35.7M ETH	Mayor participación
Seguridad económica	~$130B USD	Mantener/incrementar
Tiempo de finalidad	~12.8 minutos	**Segundos**
TPS en L1	~15-30	**10,000**
TPS en L2	~1,000-5,000	**10,000,000**
Uptime	100% desde génesis	100% (mantener)
Resistencia cuántica	Ninguna	**Completa**

3.4 ¿Por Qué Lean Ethereum Necesita Transformar Gasper?

El protocolo Gasper actual, si bien revolucionario, presenta limitaciones que Lean Ethereum busca resolver:

1. **Finalidad lenta:** 12.8 minutos es inaceptable para muchos casos de uso (bridges, pagos, DeFi).
2. **Dependencia de BLS:** Las firmas BLS12-381 son vulnerables a computadoras cuánticas.
3. **Dependencia de KZG:** Los compromisos KZG para blobs (EIP-4844) también son vulnerables.
4. **Complejidad del protocolo:** La interacción entre LMD-GHOST y Casper FFG crea vectores de ataque sutiles (balancing, bouncing, avalanche attacks).
5. **EVM legacy:** La EVM de 256 bits es ineficiente para generación de pruebas ZK.
6. **Requisito de 32 ETH:** Barrera alta para la descentralización de validadores.

---

4. Los Cimientos: GHOST y LMD-GHOST

4.1 El Protocolo GHOST Original

GHOST (*Greediest Heaviest Observed SubTree*) fue propuesto por **Yonatan Sompolinsky y Aviv Zohar** (2013). En la regla de la cadena más larga de Bitcoin, los bloques huérfanos representan trabajo desperdiciado. GHOST modifica la regla de selección: en lugar de la cadena más larga, se elige el **subárbol más pesado**.

```
Regla de Bitcoin (cadena más larga):

    [A]──[B]──[C]──[D]              ← Cadena ganadora (4 bloques)
         └──[E]──[F]──[G]──[H]     ← Cadena más larga (5 bloques) ← GANA

Regla GHOST (subárbol más pesado):

    [A]──[B]──[C]──[D]              ← Subárbol: 4 bloques
    │    └──[X]                    ← +1 bloque (contribuye al peso de A)
    │    └──[Y]                    ← +1 bloque (contribuye al peso de A)
    └──[E]──[F]──[G]──[H]          ← Subárbol: 4 bloques

    Bajo GHOST, la rama de B tiene peso 6 (B,C,D,X,Y + referencia)
    vs. rama de E con peso 4. La rama de B GANA.
```

4.2 LMD-GHOST: Adaptación para Proof-of-Stake

La adaptación de GHOST para Ethereum PoS introduce dos modificaciones fundamentales:

1. **Votos en vez de bloques:** El peso proviene de **attestaciones** de validadores (ponderadas por stake), no de bloques minados.
2. **Latest Message Driven (LMD):** Solo se cuenta el **voto más reciente** de cada validador, previniendo la amplificación de influencia.

Algoritmo Formal

```
función LMD_GHOST(store):
    // PUNTO CLAVE: Inicia desde el checkpoint justificado por Casper FFG
    bloque = store.justified_checkpoint.root

    mientras bloque tenga hijos:
        hijos = obtener_hijos(bloque)
        bloque = argmax(hijos, clave=λc. (peso(c, store), hash(c)))
    retornar bloque

función peso(bloque, store):
    w = 0
```

```
    para cada validador v en validadores_activos:
        ultimo_voto = store.latest_messages[v]
        si es_ancestro(bloque, ultimo_voto.block_root):
            w += balance_efectivo(v)
    retornar w
```

Propiedades Clave

- **Vivacidad (*Liveness*):** La cadena siempre avanza.
- **Dinamismo:** La cabeza canónica cambia con nuevas attestaciones.
- **Sin finalidad intrínseca:** Un bloque puede ser revertido si cambian los votos. → **Por eso se necesita Casper FFG.**

4.3 Lo que Lean Consensus Cambia de LMD-GHOST

Bajo Lean Ethereum, LMD-GHOST como regla de fork choice separada podría volverse **innecesaria**. Con finalidad en segundos (3-slot finality), el protocolo se asemeja más a un BFT clásico donde la selección de cadena y la finalidad se unifican en un solo mecanismo, eliminando la complejidad de la interacción dual GHOST/Casper que genera ataques como el balancing attack.

---

5. Los Cimientos: Casper FFG

5.1 Diseño y Filosofía

Casper FFG (*Friendly Finality Gadget*) fue propuesto por **Vitalik Buterin y Virgil Griffith** (2017). No es un protocolo de consenso completo, sino un **gadget de finalidad** que se superpone sobre cualquier mecanismo de producción de bloques para añadir **finalidad económica determinista**.

5.2 Checkpoints y Votos

Casper FFG opera sobre **checkpoints** (el primer bloque de cada época, cada 32 slots = 384 segundos). Los validadores emiten **votos FFG** dentro de sus attestaciones:

```
voto_FFG = (source, target)

source = Checkpoint justificado más reciente conocido
target = Checkpoint de la época actual
```

5.3 Supermayoría, Justificación y Finalización

Enlace de supermayoría `A → B` : Se establece cuando ≥ 2/3 del balance total activo vota con `source=A, target=B` .

Justificación: Un checkpoint `C` es justificado si existe un enlace de supermayoría desde un checkpoint previamente justificado hacia `C` .

Finalización (regla k=1, caso normal):

```
Sea B un checkpoint justificado en época n.
Si existe un enlace de supermayoría B → C donde C está en época n+1:
    → B es FINALIZADO (irreversible)

Diagrama:
Época n        Época n+1
  [B]─────sup──→──[C]
  justificado      justificado
  ↓
  FINALIZADO
```

Bajo condiciones normales: **finalización cada ~12.8 minutos (2 épocas).**

Bajo Lean Consensus: finalización cada ~12 segundos (3 slots).

5.4 Los Mandamientos de Casper (Condiciones de Slashing)

Dos reglas que un validador **NO debe violar**:

Mandamiento 1: No Double Vote

```
Un validador NO debe emitir dos votos distintos para la misma época target.

∀ validador v: NO ∃ a₁, a₂ tal que
    a₁ ≠ a₂ ∧ a₁.target.epoch == a₂.target.epoch
```

Mandamiento 2: No Surround Vote

```
Un validador NO debe emitir un voto que "envuelva" otro voto previo.

∀ validador v: NO ∃ a₁, a₂ tal que
    a₁.source.epoch < a₂.source.epoch < a₂.target.epoch < a₁.target.epoch

Visualización:
a₁.source ←————————————————→ a₁.target    (voto exterior)
              a₂.source ←——→ a₂.target     (voto interior)
                                            → SLASHEABLE
```

5.5 Teorema de Seguridad Responsable (Accountable Safety)

Teorema: Si dos checkpoints conflictivos son ambos finalizados, entonces **al menos 1/3 del stake total** ha violado una condición de slashing y puede ser identificado y penalizado.

Demostración (esquema):

1. Finalizar checkpoint C_1 requiere enlace de supermayoría: ≥ 2/3 del stake.
2. Finalizar checkpoint C_2 (conflictivo) requiere otro enlace: ≥ 2/3 del stake.
3. Dado que $2/3 + 2/3 = 4/3 > 1$, la intersección contiene ≥ 1/3 de validadores.
4. Estos validadores necesariamente cometieron *double vote* o *surround vote*.

Costo de romper la seguridad (estimación 2025):

```
Stake total:      ~35,700,000 ETH
1/3 del stake:    ~11,900,000 ETH
Precio ETH:       ~$3,700 USD
Costo mínimo:     ~$44,000,000,000 USD (~44 mil millones)
```

5.6 Lo que Lean Consensus Hereda de Casper FFG

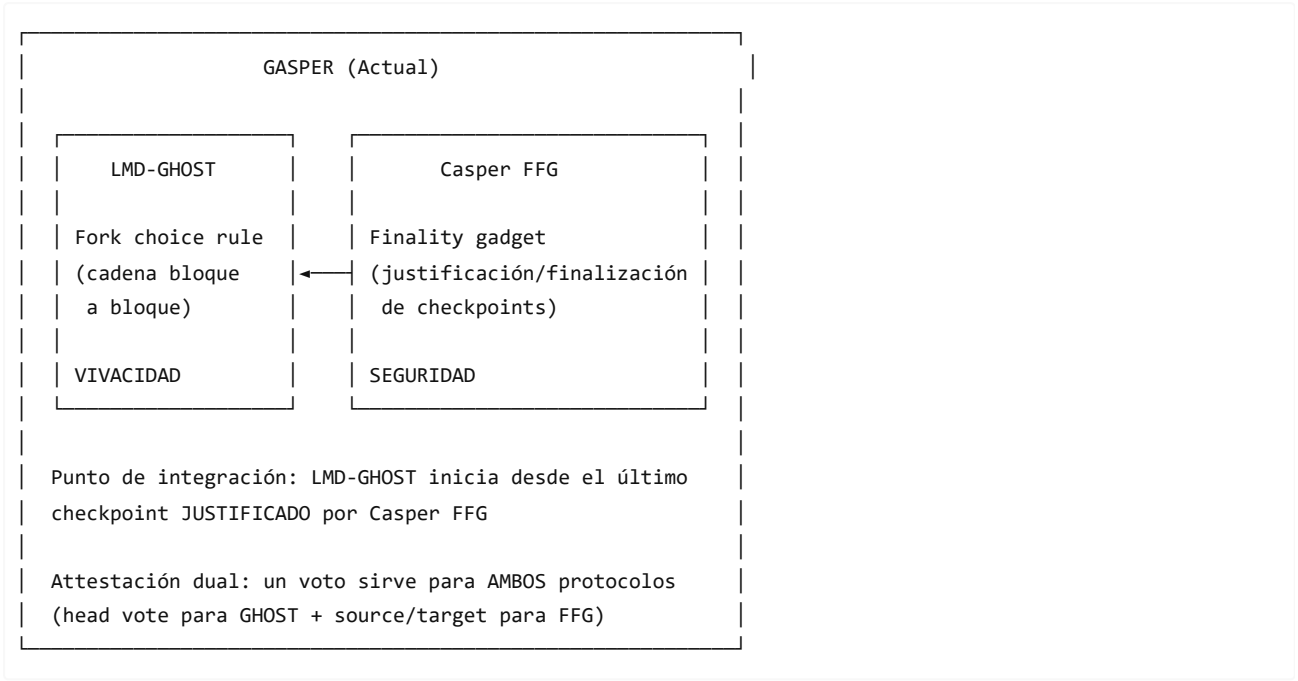
Lean Consensus **preserva la propiedad fundamental de Casper FFG** — la seguridad responsable (*accountable safety*) — pero la implementa dentro de un protocolo de finalidad mucho más rápido (3-slot finality en ~12 segundos vs. 2-epoch finality en ~12.8 minutos). Las condiciones de slashing se mantienen como garantía económica.

---

6. Gasper: El Protocolo Actual que Lean Ethereum Transforma

6.1 La Integración de Casper FFG + LMD-GHOST

Gasper — el protocolo de consenso actual de Ethereum — fue formalmente descrito en "*Combining GHOST and Casper*" (Buterin et al., 2020). El nombre es un portmanteau de **G**HOST y C**asper**.



6.2 Estructura de una Attestación (el Voto Dual)

Cada validador produce una attestación por época que sirve simultáneamente a ambos protocolos:

```
Attestation {
    slot:              uint64          // Slot asignado
    index:             uint64          // Índice del comité
    beacon_block_root: Bytes32         // ← Voto LMD-GHOST
    source:            Checkpoint      // ← Voto Casper FFG (source)
    target:            Checkpoint      // ← Voto Casper FFG (target)
    aggregation_bits:  Bitlist         // Participación del comité
    signature:         BLSSignature    // ← Firma BLS (vulnerable a cuántica)
}
```

Nota crítica para Lean Ethereum: El campo `signature` actualmente usa **BLS12-381**, que no es resistente a computadoras cuánticas. Lean Cryptography propone reemplazarlo con **firmas agregadas basadas en hashes**.

6.3 Flujo Operacional de un Slot

```
Segundo 0:   Proponente broadcast bloque (incluye execution payload)
Segundo 4:   Comités de validadores ateestan (votan por head + source/target)
Segundo 8:   Aggregators combinan attestaciones vía BLS aggregation
Segundo 12:  Inicio del siguiente slot

Cada 32 slots (1 época = 384s):
  → Se contabilizan votos FFG
  → Posible justificación/finalización de checkpoints
```

6.4 Estructura Temporal

```
1 slot  = 12 segundos
1 época = 32 slots = 384 segundos ≈ 6.4 minutos
Finalidad = 2 épocas ≈ 12.8 minutos (caso ideal)
```

6.5 Validadores en Gasper

Ciclo de vida actual:

Fase	Descripción	Duración
Depósito	32 ETH al contrato de depósito	Instantáneo
Cola de activación	Espera según churn limit (~8-12/época)	~15+ horas
Activo	Propone bloques, emite attestaciones	Indefinido
Cola de salida	Salida voluntaria	~27 horas
Retirable	Balance disponible (post-Shanghai)	Automático

Recompensas (3 componentes de attestación):

Componente	Descripción	% del máximo
Source vote	Checkpoint source correcto	~28.4%
Target vote	Checkpoint target correcto	~28.4%
Head vote	Bloque cabeza correcto	~14.2%

Fórmula de recompensa base:

```
base_reward = effective_balance / (64 × √total_active_balance)
```

Slashing (penalización por violación):

```
1. Penalización inmediata:    1/32 × balance_efectivo
2. Penalización de correlación (~18 días después):
   = 3 × (balance_slashed_periodo / balance_total) × balance_efectivo
   → Si ≥1/3 slashed simultáneamente: pérdida del 100%
3. Recompensa whistleblower:  balance_efectivo / 512
```

Inactivity Leak (auto-curación): Si no hay finalización por >4 épocas, penalización cuadrática a inactivos:

```
penalización(v) ∝ (épocas_sin_finalización)²
```

Drena validadores inactivos hasta que los activos superan 2/3 y la finalidad se reanuda.

6.6 Limitaciones de Gasper que Motivan Lean Ethereum

Limitación	Impacto	Solución Lean
Finalidad en ~12.8 min	UX pobre, riesgo de bridges	Lean Consensus: ~12 seg
Firmas BLS	Vulnerables a cuántica	Hash-based aggregate signatures
KZG commitments	Vulnerables a cuántica	Hash-based DAS commitments

Re-ejecución EVM	Ineficiente, no ZK-friendly	Hash-based real-time zkVMs
Mínimo 32 ETH	Barrera a descentralización	Reducción a 1 ETH
Complejidad GHOST+Casper	Vectores de ataque sutiles	Protocolo unificado más simple
EVM 256-bit stack	Lento para ZK proofs	RISC-V SNARK-friendly

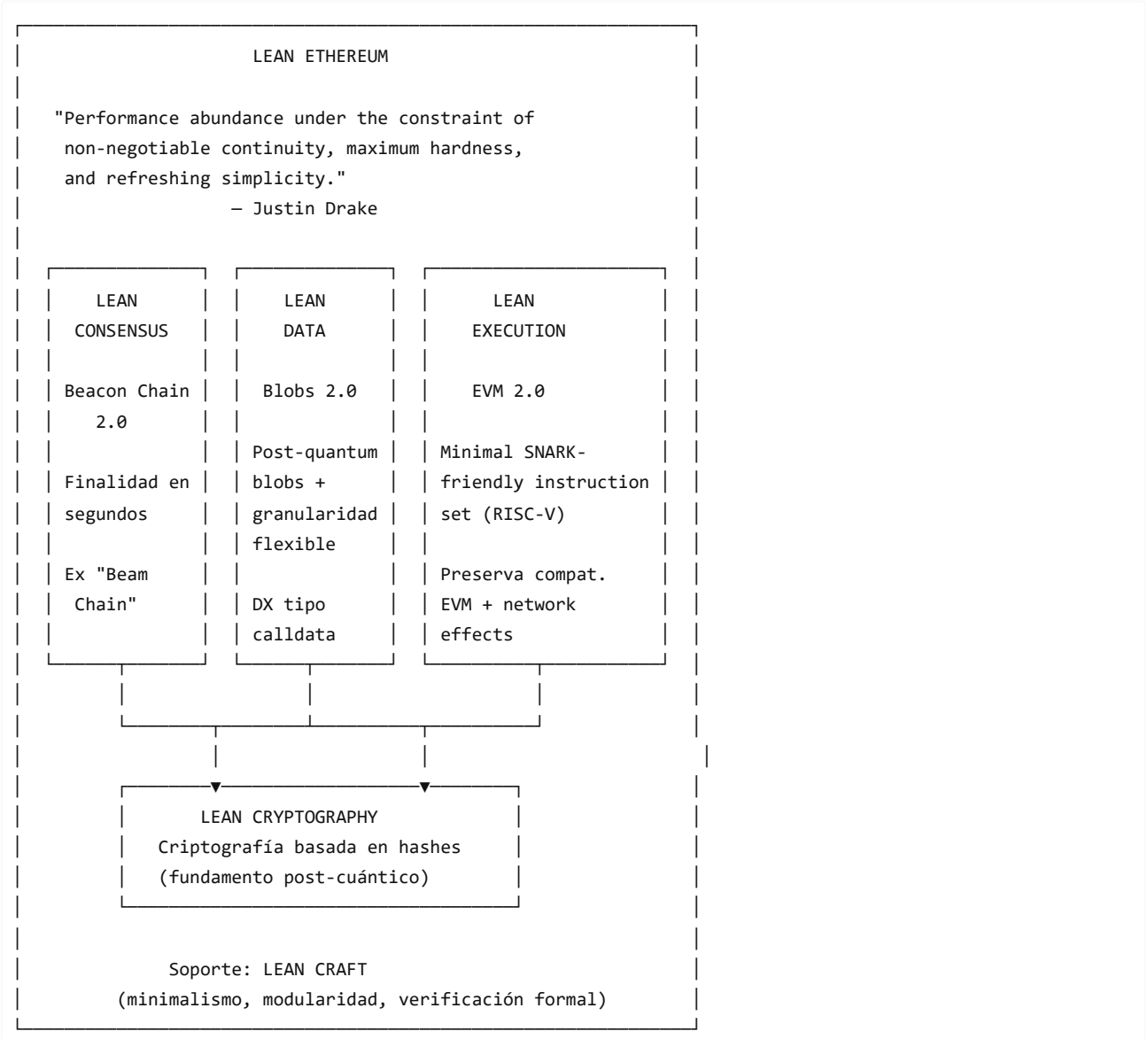
PARTE II — LA TRANSFORMACIÓN: LEAN ETHEREUM

---

7. Lean Ethereum: La Tesis Central

7.1 El Principio Unificador

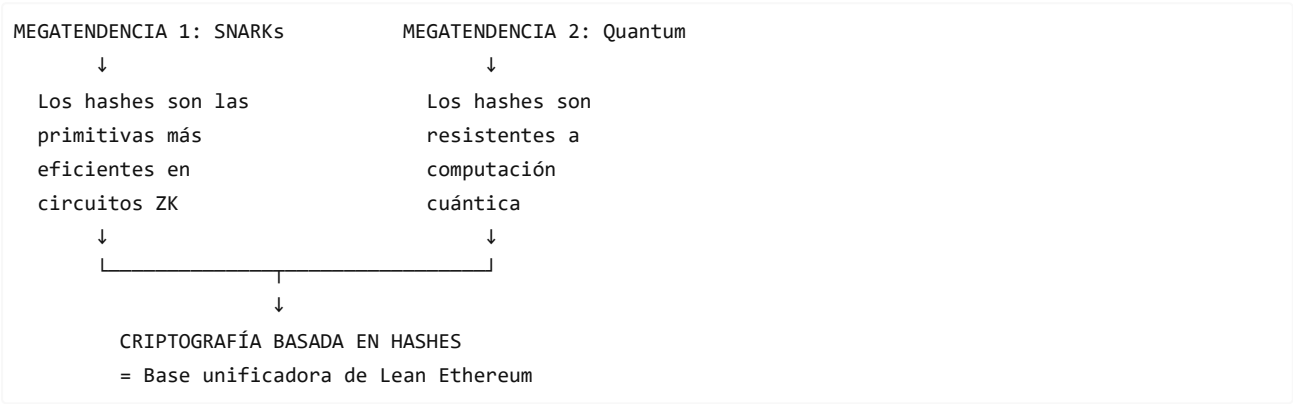
Lean Ethereum no es una colección de mejoras incrementales sino una **reimaginación desde primeros principios** de las tres subcapas de Ethereum L1, unificadas por un fundamento criptográfico común:



7.2 La Criptografía Basada en Hashes como Tesis Central

La observación clave de Drake es que la **criptografía basada en hashes** resuelve simultáneamente dos megatendencias del ecosistema:

1. **La explosión de los SNARKs:** Las pruebas de conocimiento cero (ZK proofs) están transformando cada capa de Ethereum. Los hashes son las primitivas más eficientes dentro de circuitos SNARK.
2. **La amenaza cuántica:** Las computadoras cuánticas eventualmente romperán la criptografía basada en curvas elípticas (BLS, ECDSA, KZG). Los hashes son inherentemente resistentes a ataques cuánticos.



7.3 Tabla de Transformación por Capa

Capa	Actual	Lean (Propuesto)	Cambio clave
Consenso (CL)	Firmas BLS	Firmas agregadas basadas en hashes	Post-quantum signatures
Datos (DL)	Compromisos KZG	Compromisos DAS basados en hashes	Post-quantum commitments
Ejecución (EL)	Re-ejecución EVM	zkVMs en tiempo real basadas en hashes	Verificación por pruebas ZK

---

8. Fort Mode: Defensa Inquebrantable

8.1 La Postura Defensiva

Fort Mode es la postura defensiva de Lean Ethereum. Drake argumenta que Ethereum, como *"the bedrock of the internet of value"*, debe estar preparada para sobrevivir a los adversarios más hostiles durante décadas o siglos.

8.2 Fortalezas Actuales que se Preservan

✅ 100% uptime desde el génesis (30 de julio de 2015)
✅ Diversidad de clientes inigualable (5 CL + 5 EL)
✅ $130B USD en seguridad económica (35.7M ETH × $3.7K)
✅ Accountable safety vía Casper FFG
✅ Descentralización del set de validadores (~1M+)

8.3 Amenazas que Fort Mode Neutraliza

Amenaza 1: Computadoras Cuánticas

Las computadoras cuánticas con capacidad suficiente (miles de qubits lógicos estables) podrían:

- **Romper ECDSA:** Forjar firmas de transacciones de cualquier cuenta Ethereum.
- **Romper BLS12-381:** Forjar attestaciones de validadores, comprometiendo todo el consenso.

- **Romper KZG:** Forjar pruebas de disponibilidad de datos, comprometiendo la seguridad de los blobs.

Solución Fort Mode: Reemplazar TODAS las primitivas criptográficas vulnerables con equivalentes basadas en hashes:

```
ACTUAL (Vulnerable)              LEAN (Post-Quantum)
──────────────────               ──────────────────

BLS12-381 signatures     →      Hash-based aggregate signatures
KZG polynomial commits   →      Hash-based DAS commitments
ECDSA (tx signatures)    →      Hash-based / lattice-based sigs
                                 + account abstraction
```

Amenaza 2: Estados-Nación

Un estado-nación podría intentar:

- Censurar transacciones.
- Coercer a validadores para atacar la red.
- Ejecutar ataques de largo alcance.

Defensas Fort Mode:

- **Diversidad de clientes:** Ningún cliente domina >33%, distribuyendo el riesgo.
- **Descentralización geográfica:** Validadores en múltiples jurisdicciones.
- **Resistencia a censura:** Inclusion lists (FOCIL) y encrypted mempools.
- **Inactivity leak:** Auto-curación si validadores son forzados offline.

8.4 Seguridad Económica Post-Lean

Con la reducción del mínimo de staking de 32 ETH a 1 ETH:

```
Escenario actual:  ~1M validadores × 32 ETH = ~32M ETH stakeado
Escenario Lean:    Potencialmente millones de validadores
                   Mayor descentralización
                   Mayor resistencia a coerción
```

---

9. Beast Mode: Rendimiento Extremo

9.1 Objetivos de Escalado

Beast Mode define los targets de rendimiento para la próxima década:

```
┌─────────────────────────────────────────────────────┐
│              BEAST MODE: TARGETS                    │
├─────────────────────────────────────────────────────┤
│                                                     │
│  L1 (Escalado Vertical):                            │
│  ┌──────────────────────────────────────┐           │
│  │  1 GIGAGAS / segundo                 │           │
│  │  = 1,000,000,000 gas/s               │           │
│  │  ≈ 10,000 TPS en L1                  │           │
│  │  (vs. ~15-30 TPS actuales: ~300x más)│           │
│  └──────────────────────────────────────┘           │
│                                                     │
│  L2 (Escalado Horizontal):                          │
│  ┌──────────────────────────────────────┐           │
│  │  1 TERAGAS / segundo                 │           │
```

```
│  │  = 1,000,000,000,000 gas/s              │        │
│  │  ≈ 10,000,000 TPS vía rollups L2        │        │
│  │  (1000x más que el target L1)           │        │
│  └─────────────────────────────────────────┘        │
│                                                     │
│  Tecnologías habilitadoras:                         │
│  • zkVMs en tiempo real (Lean Execution)            │
│  • Data Availability Sampling (Lean Data)           │
│  • Finalidad en segundos (Lean Consensus)           │
└─────────────────────────────────────────────────────┘
```

9.2 ¿Cómo se Logra 1 Gigagas/s en L1?

El enfoque es **vertical scaling** del L1 mediante:

1. **zkVMs:** En lugar de que cada nodo re-ejecute cada transacción, un prover genera una prueba ZK de la ejecución correcta. Los demás nodos solo verifican la prueba — órdenes de magnitud más rápido.
2. **Ejecución paralela:** Un instruction set más eficiente (RISC-V) permite mayor paralelismo.
3. **Estado optimizado:** Árboles Verkle (o binary Merkle + STARKs) reducen el overhead de acceso a estado.

9.3 ¿Cómo se Logra 1 Teragas/s en L2?

El enfoque es **horizontal scaling** vía rollups con:

1. **Blobs masivos:** Full Danksharding con DAS post-cuántico, apuntando a 32+ blobs por bloque.
2. **Data Availability Sampling:** Cada nodo solo descarga una muestra aleatoria de los datos, usando codificación de borrado para garantizar disponibilidad completa.
3. **Rollups nativos:** Los L2 publican datos en blobs y prueban ejecución correcta con SNARKs, logrando throughput masivo.

---

10. Lean Consensus: Beacon Chain 2.0

10.1 Definición

Lean Consensus es la reimaginación de la capa de consenso de Ethereum, anteriormente conocida como **Beam Chain**. Representa un "Beacon Chain 2.0" — un rediseño limpio que incorpora todas las lecciones de 5+ años de operación del Beacon Chain actual.

10.2 Comparación: Gasper Actual vs. Lean Consensus

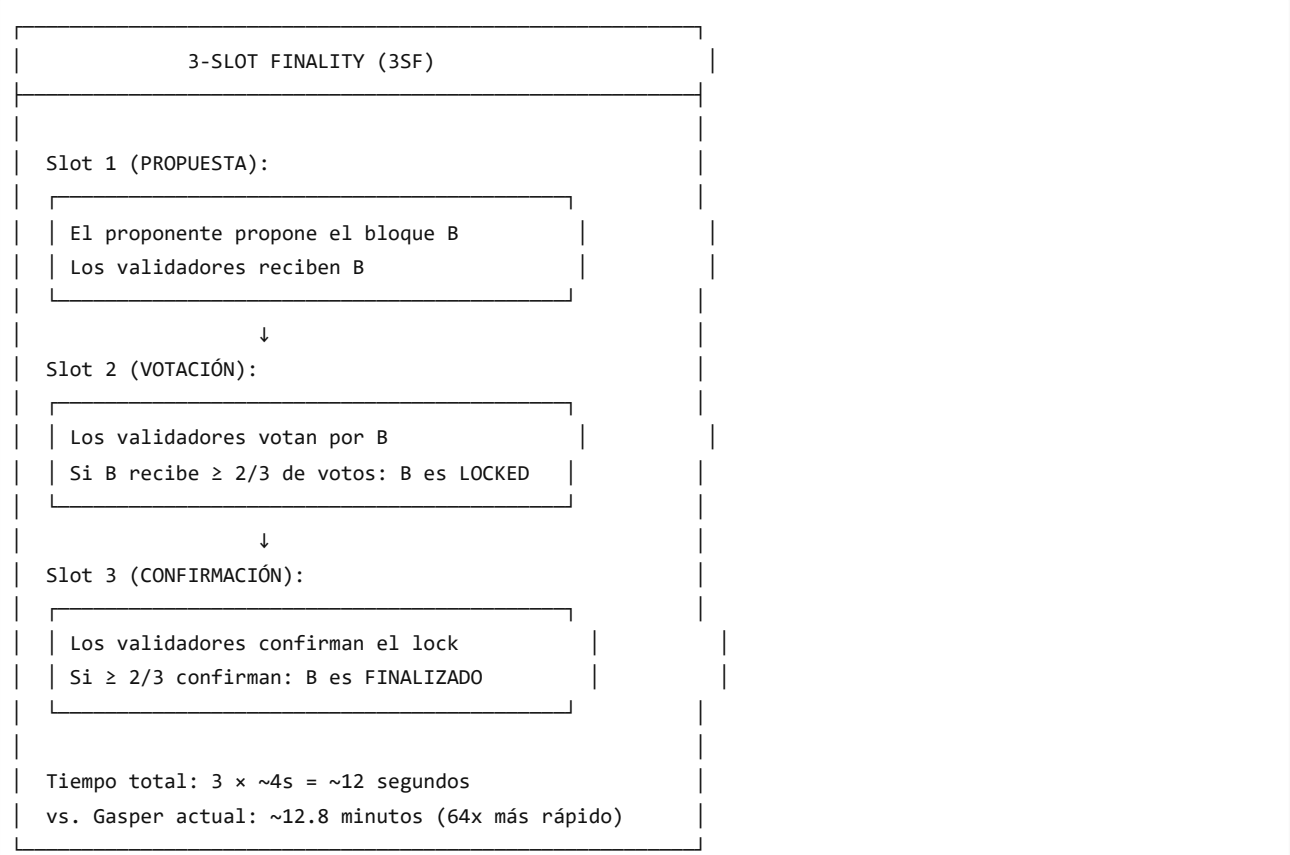
Aspecto	Gasper (Actual)	Lean Consensus (Propuesto)
Protocolo	Casper FFG + LMD-GHOST	Protocolo BFT unificado
Finalidad	~12.8 minutos (2 épocas)	**~12 segundos (3 slots)**
Block time	12 segundos	**~4 segundos**
Firmas	BLS12-381	Hash-based aggregate
Min. staking	32 ETH	**1 ETH**
Verificación	Re-ejecución completa	SNARKs del estado beacon
Aleatoriedad	RANDAO (1-bit bias)	RANDAO + VDFs
Builder separation	MEV-Boost (off-protocol)	ePBS (enshrined)
Censura resistance	Limitada	FOCIL (inclusion lists)

10.3 3-Slot Finality (3SF): Reemplazando a Gasper

La pieza central de Lean Consensus es **3-Slot Finality**, un protocolo que logra finalidad en 3 slots (~~12 segundos con slots de 4 segundos) en lugar de las 2 épocas~~ (12.8 minutos) de Gasper.

¿Cómo Funciona 3SF?

El protocolo se inspira en protocolos BFT clásicos (como Tendermint) adaptados para la escala masiva de Ethereum:



Comparación con Gasper

```
GASPER (actual):
  Slot 1...32 (época 1): Attestaciones se acumulan
  Slot 33...64 (época 2): Más attestaciones
  → Checkpoint de época 1 se FINALIZA cuando época 2 es justificada
  Total: ~768 segundos (~12.8 minutos)

3-SLOT FINALITY (Lean):
  Slot 1: Propuesta
  Slot 2: Votación (≥2/3 lock)
  Slot 3: Confirmación (≥2/3 finalize)
  Total: ~12 segundos
```

10.4 Reducción del Mínimo de Staking: De 32 ETH a 1 ETH

Lean Consensus propone reducir el mínimo de staking de 32 ETH a **1 ETH**, democratizando radicalmente la participación.

Desafío: Más validadores = más mensajes = más overhead.

Solución: La combinación de:

- **EIP-7251 (MaxEB):** Balance efectivo máximo de 2048 ETH, permitiendo consolidación.
- **Agregación de firmas basadas en hashes:** Más eficiente que BLS para sets masivos.
- **Verificación SNARK del estado beacon:** Los nodos no necesitan procesar cada attestación individualmente; verifican una prueba SNARK compacta.

10.5 Proposer-Builder Separation Nativo (ePBS)

Lean Consensus integra la separación proponente-constructor directamente en el protocolo, eliminando la dependencia de relays externos como MEV-Boost:

```
ACTUAL (off-protocol):
  Searchers → Builders → Relays (trusted) → Proposers

LEAN (enshrined PBS):
  Searchers → Builders → Protocolo (trustless) → Proposers
  + Inclusion Lists (FOCIL) para resistencia a censura
  + MEV burn (potencial)
```

10.6 FOCIL: Resistance a Censura On-Chain

Fork-Choice Enforced Inclusion Lists (FOCIL):

- Múltiples validadores (no solo el proponente) contribuyen a una lista de transacciones que **deben** ser incluidas.
- La regla de fork choice rechaza bloques que no respeten la inclusion list.
- Garantiza que ningún builder puede censurar transacciones indefinidamente.

10.7 VDFs: Aleatoriedad Mejorada

Lean Consensus propone incorporar **Verifiable Delay Functions (VDFs)** para mejorar RANDAO:

```
RANDAO actual: El último proponente de una época puede
               elegir no revelar su valor (1-bit bias attack)

RANDAO + VDF:  El valor RANDAO pasa por una VDF que toma
               tiempo fijo en calcular, eliminando la
               capacidad de manipulación
```

---

11. Lean Data: Blobs 2.0

11.1 Contexto: De Proto-Danksharding a Lean Data

EIP-4844 (Proto-Danksharding), implementado en la actualización Dencun (marzo 2024), introdujo transacciones blob (*binary large objects*) — datos de ~125 KB que se almacenan por ~18 días y luego se podan. Los blobs usan **compromisos KZG** (Kate-Zaverucha-Goldberg) para verificación.

El problema: Los compromisos KZG se basan en curvas elípticas, que son **vulnerables a computadoras cuánticas**.

11.2 Blobs Post-Cuánticos

Lean Data reemplaza los compromisos KZG con **compromisos DAS basados en hashes**:

```
ACTUAL (Proto-Danksharding):
  Blob → KZG Commitment (curva elíptica) → Verificación
  ⚠ Vulnerable a quantum
```

```
LEAN DATA:
  Blob → Hash-based DAS Commitment → Verificación
  ✅ Resistente a quantum
  ✅ Más eficiente para SNARKs
```

11.3 Granularidad Flexible de Blobs

Lean Data introduce **sizing granular** de blobs, eliminando la restricción de tamaño fijo:

```
ACTUAL:
  Cada blob ≈ 125 KB (fijo)
  Target: 3 blobs/bloque, Max: 6 blobs/bloque

LEAN DATA:
  Blobs de tamaño variable (granular)
  Developer experience similar a calldata
  Escalado hacia 32+ blobs/bloque (Full Danksharding)
```

11.4 Data Availability Sampling (DAS)

DAS es la técnica que permite escalar los datos sin que cada nodo descargue todo:

```
┌─────────────────────────────────────────────────┐
│          DATA AVAILABILITY SAMPLING             │
├─────────────────────────────────────────────────┤
│                                                 │
│  1. El blob se codifica con erasure coding      │
│     (codificación de borrado Reed-Solomon)      │
│                                                 │
│  2. Los datos se dividen en columnas            │
│                                                 │
│  3. Cada nodo descarga solo ALGUNAS columnas    │
│     aleatorias (no todo el blob)                │
│                                                 │
│  4. Si suficientes nodos muestrean exitosamente,│
│     se garantiza estadísticamente que los datos │
│     completos están disponibles                 │
│                                                 │
│  5. Cualquiera puede reconstruir los datos      │
│     completos desde un subconjunto              │
│                                                 │
│  Resultado: Throughput de datos masivo sin      │
│  aumentar requisitos por nodo                   │
│                                                 │
│  Paso intermedio: PeerDAS (2025-2026)           │
│  Paso final: Full DAS con compromisos hash-based│
└─────────────────────────────────────────────────┘
```

---

12. Lean Execution: EVM 2.0

12.1 El Problema con la EVM Actual

La Ethereum Virtual Machine actual es una máquina de stack de 256 bits diseñada en 2014. Sus limitaciones para el mundo Lean:

- **No es SNARK-friendly:** Generar pruebas ZK de ejecución EVM es extremadamente costoso computacionalmente.
- **Instruction set complejo:** Muchos opcodes legacy, gas metering complejo.
- **Overhead de re-ejecución:** Cada nodo debe re-ejecutar cada transacción para verificar el estado.

12.2 La Propuesta: RISC-V como Base

Lean Execution propone un set de instrucciones **mínimo y SNARK-friendly**, con RISC-V como candidato principal:

```
EVM ACTUAL                          LEAN EXECUTION (RISC-V)
──────────                          ──────────────────────

Stack-based (256-bit)       →       Register-based (32/64-bit)
~140 opcodes                →       Set mínimo de instrucciones
Gas metering complejo       →       Metering simplificado
ZK proof: ~minutos/tx       →       ZK proof: tiempo real
Re-ejecución obligatoria    →       Verificación vía SNARK
```

¿Por Qué RISC-V?

1. **Abierto:** Arquitectura open-source, sin licencias.
2. **Simple:** Set de instrucciones reducido y ortogonal.
3. **SNARK-friendly:** Cada instrucción RISC-V se convierte en constraints simples y predecibles dentro de circuitos ZK, reduciendo drásticamente el tiempo de generación de pruebas.
4. **Madurez:** Ecosistema extenso de compiladores, herramientas y hardware.

12.3 zkVMs en Tiempo Real

El objetivo final de Lean Execution es que la verificación de la ejecución de transacciones se haga mediante **zkVMs** (Zero-Knowledge Virtual Machines) en tiempo real:

```
MODELO ACTUAL:
  Transacción → Todos los nodos la RE-EJECUTAN → Verificación

MODELO LEAN:
  Transacción → Un prover la ejecuta y genera SNARK proof
              → Todos los nodos VERIFICAN el proof (ultra-rápido)

  Analogía: En vez de que cada nodo "haga la tarea",
  un nodo la hace y los demás verifican "la respuesta"
  con una prueba matemática irrefutable.
```

12.4 Preservación de Compatibilidad EVM

Un principio fundamental de Lean Execution es **preservar la compatibilidad con la EVM existente y sus network effects**. Los contratos inteligentes existentes y las herramientas de desarrollo deben seguir funcionando. Esto se logra mediante:

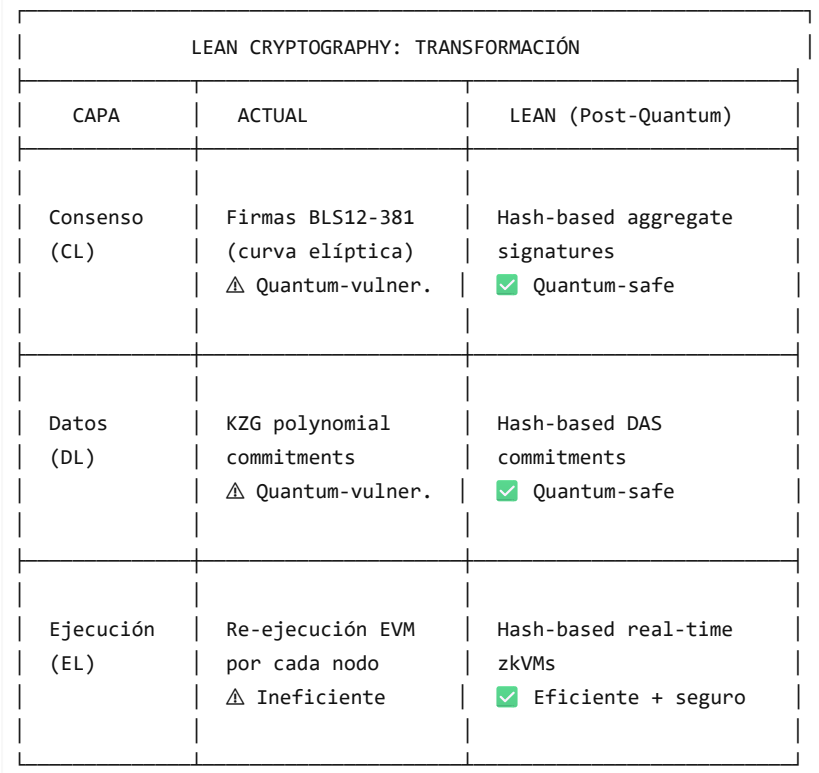
- **Compilación EVM → RISC-V:** Los contratos EVM se compilan/interpretan sobre RISC-V.
- **Abstracción de la capa de ejecución:** Los desarrolladores pueden seguir escribiendo en Solidity/Vyper.
- **Transición gradual:** No es un cambio abrupto sino una migración planificada.

---

13. Lean Cryptography: La Base Unificadora

13.1 La Transformación Criptográfica Completa

Lean Cryptography es el **fundamento transversal** que habilita las tres capas de Lean Ethereum. La tesis es simple y poderosa: reemplazar todas las primitivas criptográficas vulnerables con equivalentes basadas en hashes.



13.2 ¿Por Qué Hashes y No Lattice-Based?

Existen múltiples familias de criptografía post-cuántica (lattice-based como Dilithium/Kyber, code-based, isogeny-based). Drake argumenta a favor de **hash-based** porque:

Criterio	Hash-based	Lattice-based
Seguridad	Basada en propiedades de funciones hash, estudiadas por décadas	Basada en problemas de lattice, relativamente nuevos
SNARK-friendliness	Excelente (hashes son la primitiva nativa de SNARKs)	Moderada
Tamaño de firma	Más grande	Más compacto
Simplicidad	Muy simple	Más complejo
Madurez	Décadas de análisis criptográfico	Años
Resistencia cuántica	Probada (reducción a preimage/collision resistance)	Conjeturada

La apuesta de Lean Ethereum es que la **simplicidad y la confianza en supuestos mínimos** (solo necesitas que la función hash sea segura) superan las ventajas de tamaño de otras familias.

13.3 Firmas Agregadas Basadas en Hashes

En el consenso actual, la **agregación de firmas BLS** es fundamental: permite combinar ~1M de firmas en una sola firma verificable. Lean Cryptography necesita replicar esta capacidad con hashes.

Enfoques en investigación:

1. **SNARK-based signature aggregation:** Un prover genera un SNARK que demuestra que N validadores firmaron correctamente, comprimiendo todo a una prueba de tamaño constante.
2. **Merkle-tree based aggregation:** Combinar firmas hash-based usando árboles Merkle para agregación logarítmica.
3. **Esquemas híbridos:** Usar SNARKs para comprimir firmas basadas en hashes de forma recursiva.

13.4 Compromisos DAS Basados en Hashes

Reemplazo de KZG para data availability:

```
KZG (actual):
  Datos → Polynomial commitment → Evaluación en puntos → Verificación
  Basado en: pairing-friendly elliptic curves
  Supuesto: Discrete Log Problem (vulnerable a quantum)

Hash-based DAS (Lean):
  Datos → Erasure coding → Merkle tree commitment → Verificación
  Basado en: funciones hash (SHA-256, Poseidon, etc.)
  Supuesto: Collision resistance (quantum-safe)
```

13.5 zkVMs Basadas en Hashes

En Lean Execution, los zkVMs usan funciones hash como primitiva principal para:

- **Compromisos:** Los compromisos de estado usan Merkle trees basados en hashes.
- **Funciones de permutación:** Hash functions como Poseidon, diseñadas específicamente para eficiencia dentro de circuitos SNARK.
- **Verificación:** Los verificadores solo necesitan evaluar hashes, operación extremadamente eficiente.

---

14. Lean Craft: Filosofía de Diseño

14.1 Principios

Lean Ethereum no es solo una propuesta técnica sino una **filosofía de ingeniería**. Drake articula los principios bajo el concepto de *"Lean Craft"*:

```
┌─────────────────────────────────────────────────────────────┐
│                      LEAN CRAFT                             │
│          "When we can go the extra mile, we do."            │
│            (Analogía: "Dreams of Sushi")                    │
├─────────────────────────────────────────────────────────────┤
│                                                             │
│  ✦ MINIMALISMO                                              │
│    Cada componente justifica su existencia.                 │
│    Si algo puede eliminarse sin perder funcionalidad,       │
│    se elimina.                                              │
│                                                             │
│  ✦ MODULARIDAD                                              │
│    Las tres capas (consenso, datos, ejecución) son          │
│    independientes y pueden evolucionar por separado.        │
│                                                             │
│  ✦ COMPLEJIDAD ENCAPSULADA                                  │
│    La complejidad inevitable se encapsula en módulos        │
│    bien definidos con interfaces simples.                   │
│                                                             │
│  ✦ VERIFICACIÓN FORMAL                                      │
```

```
│    El protocolo debe ser verificable formalmente.           │
│    Propiedades como accountable safety y plausible          │
│    liveness deben ser demostrables matemáticamente.         │
│                                                             │
│  ✦ SEGURIDAD DEMOSTRABLE (Provable Security)                │
│    Las primitivas criptográficas deben tener reducciones    │
│    de seguridad formales a supuestos bien estudiados.       │
│                                                             │
│  ✦ OPTIMALIDAD DEMOSTRABLE (Provable Optimality)            │
│    Donde sea posible, demostrar que el diseño es óptimo     │
│    o cercano al óptimo dentro de su clase.                  │
│                                                             │
└─────────────────────────────────────────────────────────────┘
```

14.2 Contraste con el Enfoque Incremental

```
ENFOQUE INCREMENTAL (pre-Lean):
  Beacon Chain → parche 1 → parche 2 → ... → parche N
  Resultado: acumulación de complejidad y deuda técnica

ENFOQUE LEAN:
  Reimaginar desde primeros principios
  Diseño limpio con todo lo aprendido
  → Protocolo más simple, más seguro, más verificable
```

14.3 Verificación Formal y Lean

El nombre "Lean" también resuena con **Lean 4**, un asistente de pruebas (theorem prover) desarrollado por Microsoft Research. La visión de Lean Craft incluye la posibilidad de especificar y verificar formalmente las propiedades del protocolo en herramientas como Lean 4, Isabelle/HOL o Coq:

- **Accountable safety:** Demostrar formalmente que conflictos de finalización requieren ≥1/3 slashing.
- **Plausible liveness:** Demostrar que el protocolo siempre puede progresar.
- **Corrección del fork choice:** Demostrar que la regla de selección es consistente.
- **Soundness de los SNARKs:** Verificar que las pruebas ZK del sistema son correctas.

---

15. Ataques Conocidos al Sistema Actual y Cómo Lean los Resuelve

15.1 Tabla Comparativa

Ataque	Descripción	Gasper (Actual)	Lean Ethereum
Balancing Attack	Adversario balancea peso entre dos forks para prevenir finalización	Mitigado parcialmente con Proposer Boost (40%)	**Eliminado:** 3SF no depende de LMD-GHOST como fork choice separado
Bouncing Attack	Checkpoint justificado "rebota" entre forks, previniendo finalización	Mitigado con fork choice actualizado	**Eliminado:** Finalidad en 3 slots no permite rebotes entre épocas
Avalanche Attack	Cascada de forks que amplifican inestabilidad	Mitigado con Proposer Boost + peso por slot	**Eliminado:** Sin epoch boundary, sin acumulación de forks

Ex-ante Reorg	Proponente n+1 reorganiza bloque de slot n	Mitigado parcialmente con Proposer Boost	**Reducido drásticamente:** Finalidad en segundos minimiza ventana de reorg
Long-Range Attack	Atacante con stake histórico crea fork desde el pasado	Weak subjectivity checkpoints	**Persistente pero mitigado:** Checkpoints más frecuentes (cada ~12s)
Quantum Attack	Computadora cuántica forja firmas BLS/ECDSA	**SIN MITIGACIÓN**	**Eliminado:** Criptografía hash-based post-quantum
1-bit RANDAO bias	Último proponente manipula aleatoriedad	Impacto bajo, aceptado	**Eliminado:** VDFs
MEV censura	Builders censuran transacciones	Solo vía MEV-Boost (off-protocol)	**Mitigado:** ePBS + FOCIL (inclusion lists)

15.2 Análisis Detallado: Por Qué 3SF Elimina los Ataques de Gasper

Los ataques de balanceo, rebote y avalancha explotan la **dualidad temporal** de Gasper: los 12 segundos de cada slot (LMD-GHOST) vs. los 384 segundos de cada época (Casper FFG). Esta brecha temporal de 32x crea una ventana donde un atacante puede manipular votos antes de que la finalidad "atrape" a la selección de cadena.

Con 3SF, esta brecha **desaparece**: la finalidad ocurre cada 3 slots (~12 segundos), eliminando la ventana de manipulación:

```
GASPER: Ventana de ataque = 64 slots (2 épocas)
        entre selección de cadena y finalidad

3SF:    Ventana de ataque = 3 slots (~12 segundos)
        Insuficiente para ataques de balanceo/rebote
```

---

16. Roadmap y Timeline

16.1 Línea Temporal Propuesta

```
2022 ═══ The Merge (Sep 15) ═══ PoW → PoS
  │
2023 ═══ Shanghai/Capella (Abr 12) ═══ Retiros habilitados
  │
2024 ═══ Dencun (Mar 13) ═══ Proto-danksharding (EIP-4844)
  │      Beam Chain propuesta (Nov) ═══ Rediseño del CL
  │
2025 ═══ Pectra ═══ MaxEB (2048 ETH), account abstraction
  │      Lean Ethereum publicado (Jul 31) ═══ Visión completa
  │      Especificación ═══ Definición técnica formal
  │
2026 ═══ Desarrollo ═══ Implementación en clientes
  │      PeerDAS ═══ Data Availability Sampling parcial
  │      Preparación post-quantum ═══ Priorización de tareas
  │
2027 ═══ Testing ═══ Devnets y testnets para Lean components
  │
2028+ ═ Deploys graduales ═══ Lean Consensus, Data, Execution
```

```
  │
2029+ ═══ Visión completa ═══ Protocolo Lean operacional
  │
2035  ═══ Madurez ═══ Lean Ethereum en su forma final
```

16.2 Actualizaciones ya Implementadas que Pavimentan Lean

Dencun (Marzo 2024) — Fundamento de Lean Data

EIP	Contribución a Lean
EIP-4844 (Proto-Danksharding)	Introduce blobs y KZG, que Lean Data reemplazará con hash-based DAS
EIP-4788 (Beacon root en EVM)	Acceso trustless al estado de consenso desde ejecución
EIP-6780 (SELFDESTRUCT reducido)	Simplificación del EVM, alineado con Lean Craft

Pectra (2025) — Fundamento de Lean Consensus

EIP	Contribución a Lean
EIP-7251 (MaxEB → 2048 ETH)	Permite consolidación de validadores, prerequisito para 3SF
EIP-7549 (Committee index fuera de attestation)	Mejora eficiencia de agregación
EIP-6110 (Depósitos on-chain)	Simplificación CL, alineado con Lean Craft
EIP-7002 (Exits desde EL)	Flexibilidad de staking, paso hacia min. 1 ETH
EIP-7702 (Account abstraction parcial)	Habilita migración gradual a firmas post-quantum

16.3 leanroadmap.org

Drake lanzó **leanroadmap.org** como un tracker público de investigación, señalando un esfuerzo más transparente para organizar esta travesía multi-año. El sitio documenta el progreso técnico y organiza community calls para coordinación.

Contacto: lean@ethereum.org

---

17. Conclusiones: Lean Legacy

17.1 Síntesis

Lean Ethereum representa la visión más ambiciosa y comprehensiva para la evolución de Ethereum desde The Merge. Su tesis central es elegantemente simple: **la criptografía basada en hashes es el fundamento unificador** que permite simultáneamente resistencia post-cuántica y rendimiento extremo vía SNARKs.

El protocolo de consenso actual — Gasper, basado en Casper FFG y LMD-GHOST — fue un logro monumental que habilitó la transición a Proof-of-Stake con seguridad responsable y vivacidad robusta. Lean Consensus lo transforma, preservando las garantías fundamentales (accountable safety, slashing conditions) mientras reduce drásticamente la latencia de finalidad y simplifica la arquitectura.

17.2 Las Tres Transformaciones

```
ANTES (Gasper)                          DESPUÉS (Lean)
──────────────                          ──────────────

Consenso: Casper FFG + LMD-GHOST  →     Protocolo 3SF unificado
```

```
          Finalidad ~12.8 min          Finalidad ~12 seg
          Firmas BLS                   Firmas hash-based
          Mínimo 32 ETH                Mínimo 1 ETH

Datos:    Proto-danksharding      →    Full DAS post-quantum
          KZG commitments              Hash-based commitments
          6 blobs/bloque max           32+ blobs/bloque

Ejecución: EVM (re-ejecución)     →    RISC-V zkVM (verificación)
           256-bit stack               32/64-bit registers
           ~15 TPS                     ~10,000 TPS
```

17.3 La Promesa Generacional

Drake enmarca Lean Ethereum como **"a generational oath"** — un compromiso generacional de mantener la continuidad operacional de Ethereum, habilitar escalado sin compromisos, y asegurar que el protocolo sea digno de las generaciones futuras.

> *"If the world is online, the world is onchain."*

La declaración refleja la ambición fundamental: Ethereum como el **bedrock del internet del valor**, asegurando cientos de trillones de dólares a lo largo de décadas o siglos, resistente a estados-nación, computadoras cuánticas y cualquier adversario futuro.

17.4 Perspectiva Crítica

Es importante notar, como el propio Drake reconoce, que Lean Ethereum es una *"Drake take™"* — una visión personal, no una decisión institucional. Una diversidad saludable de perspectivas dentro de la Ethereum Foundation y la comunidad es esperada. Los desafíos de implementación son enormes:

- La migración criptográfica completa (BLS → hash-based) es sin precedentes a esta escala.
- El diseño de 3SF para ~1M validadores es un problema abierto de investigación.
- La transición de EVM a RISC-V debe preservar compatibilidad con miles de contratos desplegados.
- El timeline de una década requiere coordinación sostenida de una comunidad descentralizada.

Sin embargo, la trayectoria de Ethereum — desde la concepción de PoS en 2013 hasta su ejecución exitosa en 2022 — demuestra que esta comunidad es capaz de lograr lo que otros consideran imposible.

---

18. Referencias

Fuente Primaria

1. **Drake, J.** (2025). *lean Ethereum*. Ethereum Foundation Blog, 31 de julio de 2025. https://blog.ethereum.org/2025/07/31/lean-ethereum

Papers Académicos Fundamentales

2. **Buterin, V., Hernandez, D., Kamphefner, T., Pham, K., Qiao, Z., Ryan, D., Sin, J., Wang, Y., & Zhang, Y. X.** (2020). *Combining GHOST and Casper*. arXiv:2003.03052.

3. **Buterin, V., & Griffith, V.** (2017). *Casper the Friendly Finality Gadget*. arXiv:1710.09437.

4. **Sompolinsky, Y., & Zohar, A.** (2013). *Accelerating Bitcoin's Transaction Processing: Fast Money Grows on Trees, Not Chains*. IACR Cryptology ePrint Archive.

5. **D'Amato, F., & Zanolini, L.** (2023). *A Simple Single Slot Finality Protocol*. IACR ePrint 2023/280.

6. **Neu, J., Tas, E. N., & Tse, D.** (2021). *Ebb-and-Flow Protocols: A Resolution of the Availability-Finality Dilemma*. IEEE S&P.

7. **Schwarz-Schilling, C., Neu, J., Monnot, B., Asgaonkar, A., Tas, E. N., & Tse, D.** (2022). *Three Attacks on Proof-of-Stake Ethereum*. Financial Cryptography and Data Security.

Artículos y Recursos

8. **Blockworks.** *From Beam to Lean: Ethereum plots a century of resilience*. https://blockworks.co/news/beam-to-lean-ethereum
9. **The Block.** *EF's Justin Drake shares 10-year 'lean Ethereum' vision*. https://www.theblock.co/post/365116
10. **CryptoSlate.** *Justin Drake reveals 10-year 'Lean Ethereum' roadmap to achieve 10k TPS on mainnet*. https://cryptoslate.com/justin-drake-reveals-10-year-lean-ethereum-roadmap-to-achieve-10k-tps-on-mainnet/
11. **Bitget News.** *From Beam Chain to Lean Ethereum: An In-depth Analysis*. https://www.bitget.com/news/detail/12560604974136
12. **Bitget News.** *The Next Decade of Ethereum: Comprehensive Upgrades from Beam Chain to Lean Ethereum*. https://www.bitget.com/news/detail/12560604976520
13. **Yellow Research.** *Quantum-Proofing Ethereum: The Lean Blockchain Revolution*. https://yellow.com/research/quantum-proofing-ethereum-the-lean-blockchain-revolution-for-a-secure-future
14. **CoinDesk.** *Ethereum's Justin Drake Unveils 'Lean' Roadmap to Fend Off Quantum Threats*. https://www.coindesk.com/tech/2025/07/31/ethereum-s-justin-drake-unveils-lean-roadmap-to-fend-off-quantum-threats

Especificaciones Técnicas

15. **Ethereum Foundation.** *Consensus Specifications*. GitHub: ethereum/consensus-specs.
16. **Ethereum.org.** *Gasper*. https://ethereum.org/developers/docs/consensus-mechanisms/pos/gasper/
17. **Ethereum.org.** *Single Slot Finality*. https://ethereum.org/roadmap/single-slot-finality/
18. **Edgington, B.** *Upgrading Ethereum*. https://eth2book.info/
19. **Lean Roadmap.** https://leanroadmap.org/

Propuesta Beam Chain (Predecesora)

20. **Drake, J.** (2024). *Beam Chain: A clean-slate redesign of the Ethereum consensus layer*. Presentación en Devcon 7, noviembre 2024.

Documento compilado en febrero de 2026. Basado en la publicación oficial de Justin Drake del 31 de julio de 2025 y en la investigación académica del protocolo de consenso de Ethereum. Este es un documento de análisis — "a healthy diversity of views across Protocol, the EF, and the broader Ethereum community is expected and welcome" (J. Drake). Contacto del proyecto: lean@ethereum.org