Sistema TUC (Tiempo Universal por Consenso) en la red e-Bitcoin: Descripción Técnica

1. Definición y alcances

1.1 TUC: Tiempo Universal por Consenso (definición operativa)

El **Tiempo Universal por Consenso (TUC)** es un sistema de cronometraje compartido y acordado dentro de la red e-Bitcoin, diseñado para servir como referencia temporal única para todas las transacciones y eventos. A diferencia de un reloj externo, cuya hora es impuesta por autoridades centrales o estándares astronómicos, TUC basa su marca temporal en el consenso de la propia red. En términos operativos, TUC ofrece un **calendario cronológico inmutable y neutral** mantenido por la red, en el cual cada segundo (u otro intervalo fijo) corresponde a un bloque TUC predefinido. Este enfoque se inspira en la noción de usar la altura de bloque como estándar temporal universal, aprovechando la naturaleza descentralizada de la blockchain para establecer un **sistema de tiempo neutral** independiente de husos horarios o relojes atómicos. En esencia, TUC es la **"hora oficial" de e-Bitcoin**, acordada por todos los nodos, que permite ordenar eventos de forma consistente y proporcionar una referencia temporal verificable para cualquier acción registrada.

Importante destacar que TUC **no es simplemente un reloj** que da la hora actual, sino una **plataforma temporal integradora**. Esto significa que opera a la vez como **agenda programable** (permite agendar eventos futuros en bloques de tiempo específicos) y como **mecanismo de sellado temporal** (timestamp) una vez que dichos eventos ocurren. Cada bloque TUC actúa como un contenedor de tiempo consensuado: reúne solicitudes y las marca con la hora consensuada de la red. Así, el sistema provee a cada transacción un sello criptográfico único vinculado a un instante TUC preciso. Esta definición operativa sitúa al TUC como **columna vertebral temporal** de e-Bitcoin, sincronizando todas las actividades en un único eje temporal.

1.2 Problemas que resuelve: doble tiempo e interpolaridad

El diseño de TUC surge para resolver dos problemas centrales en entornos distribuidos y multi-cadena: el **"doble tiempo"** y la **"interpolaridad"**. El *problema del doble tiempo* se refiere a la necesidad de manejar simultáneamente dos referencias temporales diferentes – típicamente, el tiempo civil o legal (horario humano, calendario convencional) y el tiempo de la red blockchain (medido en alturas de bloque o épocas). En ausencia de un estándar unificado, las organizaciones se ven obligadas a llevar **calendarios duales**: uno regido por las fechas y horas oficiales para contratos, impuestos y obligaciones legales, y otro definido por la cronología interna de cada blockchain para la liquidación de transacciones. Esta

bifurcación impone complejidad operativa y riesgo de descoordinación, pues se debe “vivir con dos tiempos a la vez” – el reloj civil por un lado y el tiempo de bloque por otro. TUC elimina este doble registro temporal al ofrecer un único marco de tiempo consensuado para planificar y ejecutar eventos. En e-Bitcoin, todas las partes acuerdan de antemano qué momento (bloque TUC) corresponde a cada evento, alineando así la intención (tiempo prometido) con la realización efectiva (tiempo de procesamiento) en una sola escala temporal. Esto reduce ambigüedades y disputas sobre *cuándo* ocurrió algo, ya que la respuesta siempre se da en términos de TUC, accesible universalmente.

El segundo problema, la **interpolaridad**, se refiere a la dificultad de interoperar entre múltiples redes blockchain que poseen distintos relojes o parámetros de tiempo. (Nótese que *interpolaridad* es un neologismo introducido aquí para aludir a la *interpolación* de eventos entre redes diferentes mediante un tiempo común, estrechamente ligado al concepto de interoperabilidad temporal). En entornos tradicionales, coordinar acciones entre, por ejemplo, la red Bitcoin y la red Ethereum requiere lidiar con distintos intervalos de bloque, zonas horarias de referencia y mecanismos de sincronización. Típicamente, cada blockchain opera en su propia línea temporal, lo que complica la programación de operaciones *cross-chain* (entre cadenas) de forma confiable. TUC aborda esta dificultad proporcionando una **capa temporal unificada** sobre la cual se pueden agendar eventos dirigidos a diversas redes. Esto permite “interpolar” eventos de diferentes blockchains en una sola secuencia cronológica, habilitando la interoperabilidad tecnológica a través de un tiempo unificado. Por ejemplo, e-Bitcoin puede programar una transferencia en la red Bitcoin y otra en Ethereum con referencia al mismo bloque TUC (el mismo momento consensuado), garantizando que ambas ocurran con una relación temporal bien definida. De este modo, TUC actúa como **denominador común temporal** entre redes, lo que reduce la complejidad de coordinar transacciones inter-cadena y evita desfases de tiempo entre ellas. En resumen, resolviendo el doble tiempo y ofreciendo interpolaridad, TUC facilita que sistemas dispares operen sin fricción temporal, proporcionando a todos una **referencia uniforme de *cuándo* suceden las cosas**.

1.3 Alcance dentro de e-Bitcoin: contabilidad lineal y distribución por rieles

Dentro de la arquitectura de e-Bitcoin, TUC cumple un doble rol de alcance bien delimitado. Primero, sustenta la **contabilidad lineal** de la red: esto significa que todos los eventos registrados en e-Bitcoin quedan ordenados en una única secuencia temporal sin ambigüedades ni bifurcaciones. Cada transacción o transferencia ocupa una posición definida en la línea de tiempo TUC, garantizando un libro mayor estrictamente secuencial. En e-Bitcoin no existen “relojes locales” o tiempos relativos independientes; todo se registra con respecto al reloj global por consenso. Esta contabilidad lineal unificada aporta claridad y **coherencia cronológica** absoluta al histórico de operaciones: no hay solapamientos ni divergencias en el orden temporal, lo cual es crucial para la integridad contable y para evitar condiciones de carrera o dobles contabilizaciones de un mismo hecho.

El segundo rol del TUC es habilitar la **distribución por rieles** de las transacciones a diferentes redes externas sin perder la sincronización temporal. Un **riel** representa, metafóricamente, una vía de salida que conecta e-Bitcoin con otra blockchain o sistema externo. Por ejemplo, el riel `PAY-BTC` corresponde a la ruta para transferencias que deben

materializarse en la red Bitcoin, `PAY-ETH` haría lo propio para la red Ethereum, y así sucesivamente. El sistema TUC se encarga de encauzar cada transferencia por el riel apropiado **en el momento justo** estipulado, sirviendo de *hub* temporal: todas las transferencias de todos los rieles se programan en el mismo calendario TUC y, una vez llega su turno, se reparten a los distintos destinos externos. De esta forma, e-Bitcoin opera como un **ledger maestro** que conserva la secuencia lineal global de eventos, mientras orquesta la distribución por rieles a las redes correspondientes. El alcance de TUC abarca entonces todo lo relativo al *cuándo* y *en qué orden* se efectúan las transferencias, integrando la lógica de agenda con la lógica de encaminamiento. Sin embargo, cabe aclarar que, si bien e-Bitcoin/TUC controla la temporización y envío hacia los rieles, **no controla los detalles internos de la confirmación en las redes externas** (eso queda a cargo de cada blockchain destino). No obstante, incluso para esas transferencias externas, e-Bitcoin conserva un registro primario por tiempo TUC, lo que permite trazar cualquier movimiento externo de vuelta a un punto específico de la cronología común. En síntesis, el alcance de TUC en e-Bitcoin se centra en **mantener un libro mayor temporalmente ordenado** y en **actuar como intermediario temporal** para la entrega sincronizada de pagos hacia distintos sistemas, asegurando que todos ellos compartan la misma base temporal de referencia.

1.4 Qué no es TUC: no es reloj absoluto; es agenda + sello temporal

Es importante delinear los **límites del concepto TUC**, es decir, qué no representa este sistema para evitar malentendidos. En primer lugar, TUC **no es un reloj absoluto tradicional** como lo sería, por ejemplo, el Tiempo Universal Coordinado (UTC) en el mundo físico. No pretende reemplazar los estándares horarios civiles basados en astronomía o en cronómetros atómicos, ni proporciona directamente la hora del día en formato calendario gregoriano. De hecho, aunque el calendario TUC esté alineado en duración con el calendario convencional (p.ej., dividido en años, meses, días), la función de TUC no es dar la “hora del mundo real” con precisión astronómica, sino proveer un **marco de tiempo interno y autónomo** para la red e-Bitcoin. En otras palabras, TUC marca el tiempo **según el consenso de la red**, no según una autoridad externa. Esto implica que, por diseño, puede haber ligeras desviaciones o desalineamientos respecto al tiempo civil (por ejemplo, si el calendario TUC se predefinió con 365 días exactos por año ignorando segundos intercalares). Sin embargo, esas diferencias no afectan a su cometido, ya que dentro de e-Bitcoin el TUC es la fuente de verdad temporal. Así pues, los usuarios no deben confundir TUC con un estándar horario absoluto como UTC; TUC opera en su propio contexto de consenso.

En lugar de ser un reloj absoluto, TUC se concibe más bien como la combinación de **una agenda programable y un servicio de sellado temporal**. Como *agenda*, TUC permite fijar de antemano en qué bloque temporal ocurrirá un evento, cumpliendo así la función de calendario de planificación. Los participantes pueden consultar este calendario único y reservar instantes futuros para sus transacciones (como se detallará más adelante en la planificación de solicitudes). Por otro lado, como *sello temporal* (timestamp), TUC certifica criptográficamente el momento en que efectivamente ocurrió cada evento, una vez que el “tiempo presente” alcanza al bloque programado. En ese instante, el sistema emite un registro inmutable – básicamente un **hash asociado al bloque TUC** – que prueba que la transacción X ocurrió en el segundo Y según consenso. Esta doble faceta agenda+sello

significa que TUC abarca todo el ciclo temporal de un evento: desde su **programación futura hasta su registro pasado**.

Por último, conviene destacar que TUC **no funciona como un oráculo externo** de tiempo para otras aplicaciones fuera de e-Bitcoin, salvo a través de la interoperabilidad prevista. Su propósito es intrínseco a la red: coordinar internamente y servir de puente temporal con otras blockchains en el contexto de transacciones. No debe entenderse TUC como un servicio general de timestamp público para documentos (aunque la propia red e-Bitcoin podría usarse con ese fin); su implementación está específicamente orientada a resolver la sincronización y secuenciación de transferencias dentro del ecosistema e-Bitcoin. En suma, TUC no es un reloj universal absoluto en términos físicos, **sino un sistema temporal consensuado, al servicio de la agenda transaccional y la estampación confiable de la hora en que suceden los eventos**, todo ello dentro y a través de la red e-Bitcoin.

2. Principios del sistema

2.1 Pulso constante y calendario anual predefinido

El sistema TUC se basa en un **pulso temporal constante**: una cadencia fija de bloques de tiempo que avanzan uniformemente, marcando el ritmo de la red. En la implementación de e-Bitcoin, este pulso puede concebirse, por ejemplo, como un bloque TUC por segundo (o por otro intervalo constante determinado). Lo crucial es que el intervalo es **estrictamente regular y conocido de antemano**, sin variaciones aleatorias ni ajustables por participantes. Esto contrasta con blockchains tradicionales como Bitcoin, donde los bloques ocurren en promedio cada 10 minutos pero con variación aleatoria. En TUC, la aparición de bloques es **periódica** y se rige por un calendario preestablecido: desde el inicio del año hasta el final, cada unidad de tiempo (segundo, minuto, etc.) está asignada a un bloque específico.

El **calendario anual predefinido** de TUC significa que, antes de comenzar un nuevo año (o periodo), la red ya conoce cuántos bloques contendrá ese año y cómo se distribuyen por mes, día, hora, etc. Por ejemplo, si TUC opera a resoluciones de segundos, un año no bisiesto tendría 31.536.000 bloques enumerados secuencialmente. Estos bloques pueden organizarse jerárquicamente: se sabe que hay 60 bloques por minuto, 3.600 por hora, ~86.400 por día, ~2.592.000 por mes (variable según mes) y así sucesivamente, con los ajustes pertinentes en febrero o años bisiestos según se defina. **Las plantillas de calendario** se acuerdan de antemano (ver sección 3.2), de modo que todos los nodos comparten la misma noción de la estructura temporal. Esto aporta predictibilidad total: cualquier nodo o usuario puede calcular exactamente qué identificador de bloque TUC corresponde, por ejemplo, al 10 de mayo del año en curso a las 15:45:00, sin necesidad de esperar a que ocurra. El pulso constante evita incertidumbre en la sincronización – la red late al unísono, generando “ticks” de tiempo regulares.

Este principio de pulso constante proporciona la **base para la coordinación temporal**: al saber de antemano cuándo caerá cada bloque, los participantes pueden alinear sus acciones con confianza. Además, elimina la dependencia de fuentes de tiempo externas: la red en sí misma marca los segundos a través de su consenso, funcionando como un “reloj interno colectivo”. Cabe resaltar que mantener un pulso constante requiere que algún

mecanismo de consenso (o un rol especial, e.g. el *Block Creator* mencionado más adelante) genere puntualmente cada bloque, ya sea que contenga transacciones o esté vacío. Así, incluso en ausencia de actividad, el tiempo consensuado sigue avanzando bloque a bloque sin retrasos. En resumen, la filosofía de TUC es la de un **tiempo determinístico y compartido**: todos conocen la secuencia temporal con precisión, lo que sienta las bases para la planificación robusta y elimina discrepancias debidas a variabilidad de intervalos.

2.2 Bloque TUC como unidad de contabilidad en tiempo lineal

El **bloque TUC** constituye la unidad fundamental de contabilidad del tiempo en e-Bitcoin. Cada bloque TUC corresponde a un intervalo elemental (por ejemplo, un segundo) en la línea temporal lineal del sistema, y sirve como "contenedor" donde se registran las operaciones ocurridas en ese lapso. Conceptualmente, se puede pensar el bloque TUC como un folio en un libro diario: en cada folio se anotan todas las transacciones admitidas durante ese instante, dejando así un rastro permanente de qué sucedió en cada momento. La secuencia continua de bloques TUC conforma un **eje temporal lineal**, un historial encadenado que va desde el génesis de la red hasta el presente, sin ramas ni saltos discontinuos. Esta propiedad lineal es esencial: garantiza que cualquier evento A que preceda a otro B en el tiempo siempre quedará antes de B en la cadena de bloques TUC, y viceversa.

Dado que cada bloque TUC está identificado de forma única (mediante un número de bloque o una etiqueta temporal específica), resulta sencillo referenciar cuándo ocurrió algo: basta citar el bloque TUC correspondiente. Esto proporciona una notación natural para la cronología de e-Bitcoin. Por ejemplo, se podrá decir "la transferencia X fue procesada en el bloque TUC #2026-05-10 15:45:00" si se usa un formato de fecha-hora, o en un número de serie equivalente. A diferencia de blockchains donde la noción de tiempo es un campo dentro del bloque (como la marca Unix timestamp que los mineros incluyen en Bitcoin), aquí el **bloque en sí es la representación del tiempo**: su posición en la secuencia ya indica la hora por consenso. Todos los nodos concuerdan en la numeración y orden de estos bloques, por lo que se elimina cualquier discrepancia sobre el orden temporal de las transacciones.

Al usar el bloque TUC como unidad contable, e-Bitcoin logra una **trazabilidad temporal absoluta**. Cada transacción es contabilizada no solo por sus cambios de saldo sino también por su marca de tiempo exacta dentro de la contabilidad. Esto aporta beneficios en muchos frentes: por ejemplo, en auditoría, se puede reconstruir el libro mayor sabiendo en qué bloque TUC entró cada movimiento; en cumplimiento regulatorio, se puede demostrar que un pago ocurrió antes o después de cierta fecha con la precisión de segundos consensuados; en conciliar con sistemas externos, se puede mapear la hora TUC a hora civil cuando sea necesario (p.ej., para informes financieros). En suma, el bloque TUC actúa como la "moneda temporal" en e-Bitcoin, una unidad indivisible de tiempo lineal que encapsula la información de los eventos ocurridos en ese intervalo, y cuya sucesión secuencial construye la narrativa cronológica íntegra de la red.

2.3 Transferencia como unidad universal de servicio (no gratuito)

Dentro del sistema TUC, la operación básica que los usuarios pueden realizar es la **Transferencia**, la cual se erige como la unidad universal de servicio que presta la red. Aquí "transferencia" tiene un sentido amplio: abarca todo evento que un usuario agenda o ejecuta a través de e-Bitcoin, típicamente asociado al movimiento de valor (como un pago) o a la anotación de un hecho (como un registro de documento) en un momento dado. Independientemente de si se trata de enviar fondos internamente, pagar a una dirección externa por un riel, o simplemente sellar un hash para certificar algo, todas estas acciones se formalizan como *Transferencias* dentro del sistema. Esta unificación conceptual simplifica la interfaz: desde la perspectiva de TUC, todo es una transferencia de un estado a otro, solicitada por un usuario, que ocurrirá (o ocurrió) en cierto bloque TUC.

Es importante subrayar que cada Transferencia constituye un **servicio no gratuito**. Es decir, consumir un bloque de tiempo en la agenda y procesamiento de e-Bitcoin conlleva un costo o al menos un uso de recursos limitado. Al ser una unidad de servicio, implica que hay mecanismos de cuota, tarifa o gasto de algún crédito para efectuarla, asegurando así que el sistema no sea abusado con peticiones infinitas. Esto sigue la lógica de la mayoría de las redes blockchain donde las transacciones consumen *fees* o gas: e-Bitcoin no proporciona transferencia de tiempo/valor ilimitada sin costo, sino que cada Transferencia debe *pagar* de alguna forma por el uso de la infraestructura (sea mediante una tarifa en la moneda nativa, deducción de saldo, o un boleto de acceso asignado previamente, según cómo esté implementado económicamente). El **principio de no gratuidad** cumple dos funciones: por un lado, regula la demanda (previniendo spam y saturación por uso irresponsable) y, por otro, posiblemente financia la operación de la red (remunerando a los nodos creadores de bloques o mantenedores del sistema de consenso).

Al estandarizar la Transferencia como unidad de servicio, e-Bitcoin permite también medir y gestionar la **capacidad temporal** de la red en términos de cuántas transferencias por segundo (o por bloque) puede manejar. Cada bloque TUC tiene un límite de cuántas transferencias admitir (ver sección 5.4), lo que a su vez se traduce en la capacidad transaccional. Las transferencias agendadas y en tiempo real compiten por estos espacios, y al tener asociado un costo o prioridad neutral, se garantiza un tratamiento equitativo (no hay atajos sin costo). En resumen, la Transferencia es al sistema TUC lo que una transacción es a una blockchain tradicional: la acción básica, atómica, cuyo procesamiento consume recursos y cuyo resultado es registrado. La filosofía es que **el servicio de tiempo unificado es valioso y escaso**, por lo que cada uso de él debe ser deliberado y con contraprestación, asegurando la sostenibilidad del sistema.

2.4 Transferencia = adquisición de hash de que algo sucede en un momento específico

Un aspecto esencial del modelo TUC es que cuando un usuario realiza una Transferencia, lo que realmente obtiene a cambio es un **registro inmutable (un hash criptográfico) que certifica que *algo sucedió en un momento específico***. En otras palabras, la **consumación de una Transferencia equivale a obtener un sello de tiempo por consenso** que vincula inequívocamente el evento (pago, registro, etc.) con un bloque TUC particular. Esta noción convierte a cada transferencia en una suerte de adquisición de prueba temporal: el resultado final no es solo que fondos se movieron de A a B, sino que

además queda constancia pública de *cuándo* ocurrió ese movimiento, bajo la fe notarial de la red.

Dicho hash funciona como una **huella digital temporal** del evento. Usualmente, en cada bloque de una blockchain se calcula una raíz de Merkle o un hash global que agrupa todas las transacciones confirmadas en ese bloque. e-Bitcoin no es la excepción: al procesarse una transferencia en el bloque TUC X, formará parte del cálculo criptográfico que produce el hash identificador de ese bloque TUC. Ese hash de bloque, o incluso la transacción individual si se indexa, es la prueba de que la acción tuvo lugar a la hora X según consenso. Todos los participantes acuerdan esa marca de tiempo, lo que confiere a la transferencia un certificado de temporalidad. En términos de confianza, es análogo a un notario que estampa fecha y hora en un documento, salvo que aquí es la red descentralizada la que actúa de notario colectivo. Una *marca de tiempo de consenso* aporta garantías de ordenación y evita ambigüedades: al haber una hora verificada para cada transacción, todos los usuarios confían en el orden cronológico de los eventos registrados. Esta práctica, utilizada en blockchain para prevenir problemas como el doble gasto mediante la secuencia temporal, se lleva en TUC al máximo nivel de precisión (resolución de segundos o la unidad mínima definida).

En consecuencia, cuando alguien utiliza el sistema TUC, en el fondo está **adquiriendo certeza temporal**. Por ejemplo, si se agenda el pago de un alquiler para el bloque TUC del 1° de mes a las 00:00:00, tanto pagador como receptor obtienen, una vez procesado, una evidencia incuestionable (el hash/sello del bloque correspondiente) de que el pago efectivamente se ejecutó en ese momento preciso. Esta evidencia puede verificarse independientemente: cualquier tercero puede comprobar, a través de los datos de la cadena e-Bitcoin, que la transacción aparece en el bloque con timestamp consensuado tal. Así, la Transferencia funciona como un **servicio de timestamping financiero**: garantiza no solo la transferencia de valor sino su inscripción en el tiempo. Esto cobra enorme relevancia en escenarios de auditoría, contratos programados y acuerdos legales, porque elimina disputas sobre el momento de cumplimiento. De hecho, se puede decir que **el valor intrínseco de una transferencia TUC es su marca temporal**: mover dinero de A a B también podría lograrse por otros medios, pero moverlo con un sello de tiempo certificado por la red es lo que distingue al servicio. En síntesis, cada Transferencia en e-Bitcoin con TUC se convierte en un hecho con **validez temporal universal**, respaldado criptográficamente, que el usuario adquiere y puede presentar como prueba de “esto sucedió en tal segundo, según el consenso de todos”.

3. Preparación del sistema (antes de que exista cualquier solicitud)

3.1 Block Creator: rol fundacional

Para que el sistema TUC funcione desde el arranque, se requiere un **rol fundacional** encargado de establecer y mantener el pulso temporal: el *Block Creator*. Este término se refiere a la entidad (ya sea un nodo específico, un conjunto de nodos, o un algoritmo preprogramado) responsable de generar los bloques TUC conforme al calendario previsto, especialmente en ausencia de transacciones. El Block Creator desempeña un papel similar

al de un “metrónomo” de la red e-Bitcoin, asegurando que los bloques aparecen con la cadencia exacta acordada (por ejemplo, cada segundo). En el génesis del sistema, es el Block Creator quien inicia la secuencia temporal creando el bloque TUC cero y los subsiguientes en orden. Podemos imaginar que, antes de que haya solicitudes de usuarios, este rol produce bloques vacíos (o con información mínima) siguiendo el cronograma – simplemente para que el tiempo consensuado avance.

El Block Creator tiene un carácter *fundacional* porque, al menos en la fase de arranque o establecimiento del calendario, actúa con cierta autoridad para poner en marcha la infraestructura temporal. Podría tratarse de un algoritmo integrado en el protocolo (por ejemplo, en redes Proof of Stake delegadas, se podría designar un validador líder por época que emita los bloques de cada segundo). Alternativamente, podría ser un consorcio inicial que firma los bloques base del año. Sea cual sea la implementación, este rol inicia la **cadena temporal de confianza**: los primeros bloques, sellados por el Block Creator, sientan la base sobre la cual todos los nodos sincronizan su reloj interno TUC. A partir de allí, idealmente el mecanismo de consenso de e-Bitcoin (ya sea PoS, PoW adaptado, u otro) tomaría las riendas para validar y asegurar la continuidad de los bloques emitidos.

Cabe destacar que el Block Creator no necesariamente decide el contenido de las transacciones (eso viene de las solicitudes de usuarios), sino que se enfoca en la **estructura temporal y la emisión de plantillas de bloque**. En la práctica, su función podría abarcar: (a) publicar las *plantillas de calendario* para cada periodo (sección 3.2), (b) generar los identificadores de bloque y encabezados básicos de cada segundo, y (c) propagar estos bloques a la red, incluso si vacíos, para que todos los relojes sigan alineados. Una vez que los usuarios comiencen a enviar solicitudes, el Block Creator incluirá esas transacciones en los bloques que corresponde, pero siempre manteniendo la disciplina temporal. En sistemas más descentralizados, este “creador de bloques” puede rotar o ser elegido, pero en la concepción fundacional se asume que existe al menos un agente inicial que arranque la secuencia. Sin él, el TUC carecería de ancla inicial y coordinación: alguien debe dar el primer paso para que luego la red completa siga el compás. Por eso es fundacional: es la semilla que arranca el tiempo por consenso en e-Bitcoin.

3.2 Plantillas de calendario: bloques predefinidos por año

Antes de que cualquier usuario interactúe con el sistema, e-Bitcoin debe contar con una **plantilla de calendario** establecida para su tiempo de consenso. Esto significa que la estructura de bloques para un periodo extenso (típicamente un año) está **predefinida** en cuanto a su cantidad y organización. Una plantilla de calendario es esencialmente un *cronograma de bloques TUC*: detalla, por ejemplo, que el año 2026 tendrá 31.622.400 bloques (si incluye un día bisiesto, etc.), enumerados del 0 al 31.622.399, y asignados a fechas y horas concretas. Estas plantillas podrían generarse algorítmicamente (p.ej., siguiendo reglas de calendario gregoriano para asignar bloques a fechas) y ser distribuidas a todos los nodos como parámetro del protocolo. Alternativamente, podrían ser emitidas anualmente por el Block Creator o acordadas via actualización de red.

El propósito de predefinir los bloques es doble: por un lado, garantiza que **todos los participantes tengan la misma expectativa temporal** (no habrá debates sobre cuándo

termina o inicia un día en TUC, pues está especificado en la plantilla); por otro lado, permite funcionalidades como la planificación anticipada de eventos. Con la plantilla anual en mano, un usuario o contrato inteligente puede desde ya ubicar, por ejemplo, “el primer lunes de octubre a las 09:00” como un bloque TUC específico, sabiendo que existirá. Esto es crucial para casos de uso como agendar pagos mensuales recurrentes, donde quizá se quiere programar con años de antelación.

La plantilla de calendario incluye la **numeración y estructura de bloques por unidad de tiempo** (como se menciona en 3.3). Podría presentarse como una tabla que indique: en enero del año N hay 31 días * 86400 bloques por día, enumerados del X al Y; febrero tiene 28 días * 86400 bloques; etc. Cualquier ajuste necesario, como incorporar un segundo intercalar (leap second) si el diseño lo contemplara, estaría ya reflejado en la plantilla, evitando sorpresas en tiempo real. De esta forma, el calendario TUC está alineado con el calendario civil en términos de conteo de días y meses, facilitando su adopción y entendimiento por humanos (aunque internamente no dependa de relojes externos, mantiene correspondencia en nomenclatura).

La creación de estas plantillas puede considerarse parte de la configuración de génesis de e-Bitcoin. De hecho, podría haberse definido una larga secuencia de años desde el principio. Alternativamente, la red puede generar y acordar una nueva plantilla cada año (lo que exigiría un consenso anual pero no continuo). En cualquier caso, antes de que exista la primera solicitud de usuario, los bloques futuros ya están “reservados” conceptualmente en el calendario. Esto **brinda certeza temporal de largo plazo**: los participantes conocen de antemano el espacio en el que podrán operar. Por ejemplo, una empresa sabe que el 1 de enero del año próximo a las 00:00 corresponde al bloque TUC #X, y puede planificar emitir bonos o pagos en ese bloque. Sin un calendario predefinido, habría incertidumbre y la imposibilidad de planificar más allá de un horizonte corto. Por eso, la preparación del sistema incluye este paso: **fijar la agenda macro del tiempo consensuado** antes incluso de que lleguen órdenes concretas.

3.3 Numeración y estructura: bloques por minuto/hora/día/mes/año

La numeración de los bloques TUC sigue una lógica estructurada que refleja la jerarquía temporal convencional. Cada bloque puede identificarse no solo por un índice secuencial absoluto, sino también por sus componentes de tiempo: minuto, hora, día, mes, año, dentro del calendario TUC. Por ejemplo, podríamos decir “Bloque TUC 2026-05-10 15:45:00” para referirnos al bloque correspondiente a esa fecha y hora exactas (asumiendo formato AÑO-MES-DÍA hh:mm:ss). Esta nomenclatura humana se corresponde a su vez con un número de bloque dentro de la cadena. La estructura jerárquica implica que sabemos, por construcción, cuántos bloques hay por minuto (p.ej., 60 si la resolución es por segundo), por hora (~3600), por día (~86.400), por mes (variable, alrededor de 2.592.000 en un mes de 30 días), etc.

La **numeración** se puede llevar de modo absoluto (contando desde el inicio de la era TUC) o reiniciándola cada año para facilitar la lectura. Un diseño posible es: cada bloque tiene un sello con campos [año, mes, día, hora, minuto, segundo] y un número de secuencia interno. Otro diseño es numerar globalmente e inferir la fecha mediante cálculo modular. Lo importante es que la estructura es *conocida y fija*. Por ejemplo, sabiendo que cada día tiene

86.400 bloques, un cliente puede saltar en la cadena en incrementos de 86.400 para moverse día a día. Esto ayuda en indexación y búsqueda de eventos por fecha.

Esta regularidad estructural transforma la cadena TUC en algo similar a un **calendario lineal encapsulado en un ledger**. Permite también implementaciones eficientes: se pueden crear árboles o índices por cada intervalo (un árbol por años, subdividido en meses, etc.) para gestionar las solicitudes y entradas de forma ordenada. Desde la perspectiva de la red, cada bloque "sabe" en qué posición del año está, lo que puede aprovecharse para validaciones (por ejemplo, un nodo al recibir un bloque con timestamp, verifica que encaje en la secuencia y no salte un número).

En la fase de preparación del sistema, definir la numeración y estructura significa que todos los nodos configuran sus calendarios con la misma lógica. Si la red introduce un año bisiesto, todos sabrán que el bloque de la medianoche del 29 de febrero existe ese año. Si hay algún ajuste (ej: omitir segundos extra), se aplicará uniformemente en la estructura. **No hay improvisación temporal en tiempo de ejecución**; está todo delineado. Esta homogeneidad es clave para evitar bifurcaciones: imagínese que sin acuerdo previo, la mitad de la red cree que a medianoche hay un bloque extra y la otra mitad no, se produciría una inconsistencia. Por eso la estructura temporal es parte del consenso inicial.

En resumen, la numeración y estructura de los bloques TUC siguen el esquema natural de las unidades de tiempo, garantizando que por cada minuto, hora, día, etc., existe un número exacto de bloques predeterminado. Esto brinda una **navegabilidad temporal** a la blockchain e-Bitcoin, facilitando la vida tanto a los componentes automatizados (smart contracts que calculan plazos) como a los humanos (que pueden mapear mentalmente un bloque a una fecha). Es la columna vertebral organizativa que hace del TUC un calendario-lineal-digital robusto y confiable.

3.4 Agenda unificada: un solo calendario para todo

Uno de los principios rectores de TUC en e-Bitcoin es que toda la actividad se rige por **una única agenda unificada**. Esto significa que **existe un solo calendario compartido para todos los usuarios, propósitos y rieles**, sin divisiones o sub-agendas aisladas. En la práctica, cada bloque temporal es único en la red y potencialmente puede contener transacciones de cualquier tipo; no hay "doble agenda" en la cual, por ejemplo, los pagos internos sigan un calendario y las transferencias externas otro. Todo converge en la misma línea de tiempo TUC.

Esta unificación es crucial para evitar precisamente el problema de "doble tiempo" descrito previamente. Si hubiera agendas separadas (imaginemos, un calendario para contratos inteligentes y otro para pagos financieros), se volvería a incurrir en la necesidad de sincronización entre ellas. En cambio, con un único calendario global, cualquier evento de cualquier naturaleza que requiera programarse o registrarse lo hace en referencia al mismo eje temporal. Esto aporta **claridad conceptual y técnica**: no importa de qué tipo de servicio de e-Bitcoin se trate (pago de nómina, ejecución de un derivado, registro de un documento, etc.), todos consultan y utilizan la misma agenda TUC. La consecuencia es que en la red **solo puede haber un "presente" a la vez**: un único bloque es el presente en cada

momento, marcando el hito a procesar, y todo lo demás está o en el futuro (pendiente) o en el pasado (ya procesado).

La agenda unificada se manifiesta también en las interfaces hacia los usuarios: cualquiera, desde su llave (Key), ve el mismo calendario que los demás. No hay calendarios privados ni locales — por diseño, la consistencia global exige uno solo. Esto recuerda a un sistema operativo multiusuario con un único scheduler de CPU: todos los procesos compiten en la misma cola de tiempo, asegurando equidad y orden global. Análogamente, e-Bitcoin con TUC es un *scheduler* global de transacciones.

Un beneficio directo de esta unidad es la **simplificación de interoperabilidad interna**. Por ejemplo, si un contrato A programa internamente un evento para un cierto bloque y un usuario B programa un pago para ese mismo bloque, ambos eventos concurrirán en el mismo contenedor temporal, pudiendo incluso interactuar (p.ej., la ejecución del contrato A puede depender de que el pago de B esté en el mismo bloque, lo que sería posible al compartir agenda). Esta sincronía no sería factible si cada cosa tuviera su calendario separado.

La unificación requiere que en la fase de preparación todos los participantes adopten la misma fuente de tiempo (el Block Creator y plantillas ya discutidos). Una vez en marcha, **el calendario TUC es la referencia suprema**: si hay alguna incertidumbre, la resolución siempre es consultar el estado de la agenda unificada. Por ejemplo, para saber si un intervalo está libre para agendar algo, se consulta la misma agenda que todos consultan. Esto evita condiciones de carrera del tipo “reservé este tiempo pensando que estaba libre pero otro también lo reservó”, porque la reserva ocurre globalmente y de forma atómica en la agenda única.

En resumen, e-Bitcoin opera con un solo calendario TUC para todo y para todos. Esto es la esencia de su simplicidad y potencia: la diversidad de acciones y redes se armoniza bajo un **mismo compás temporal**. Los usuarios y sistemas solo tienen que acordar en una cosa – qué bloque temporal – en lugar de coordinar múltiples timelines. La agenda unificada de TUC es, por tanto, el **lenguaje común de coordinación** dentro de e-Bitcoin.

3.5 Capas solapadas y rieles: nomenclatura PAY-BTC, PAY-ETH, etc.

Aunque el calendario es único, el sistema soporta **capas solapadas** de actividad gracias a la noción de *rieles*. Un **riel** representa una capa o canal de salida específico que conecta la red e-Bitcoin con una red externa particular (o con un tipo de servicio particular). Por ejemplo, `PAY-BTC` indica el riel de pagos hacia la blockchain de Bitcoin; `PAY-ETH` el riel hacia Ethereum; podrían existir rieles `PAY-LTC` (Litecoin), `PAY-BCH`, etc., según las integraciones soportadas. Estos rieles funcionan *sobrepuestos* en el sentido de que comparten el mismo calendario TUC pero segregan las transferencias por destino.

Imaginemos las capas como transparencias superpuestas: en una misma ventana de tiempo TUC, podríamos tener transacciones internas de e-Bitcoin (capa interna) y simultáneamente transacciones en riel BTC, en riel ETH, etc., cada capa con sus destinos distintos. Todas ocurren en el mismo bloque TUC pero **etiquetadas** con su riel correspondiente. La nomenclatura `PAY-XYZ` es la forma de denotar la capa/riel de pago

hacia la red XYZ. Esta etiqueta acompaña a cada transferencia desde su programación: al agendar una transferencia, el usuario especifica el riel, por ejemplo “enviar 0.5 BTC por riel PAY-BTC al destinatario Z”. De ese modo, cuando llegue el momento, e-Bitcoin sabe que esa transferencia pertenece a la capa Bitcoin y deberá encaminarla al lugar correcto (el *lobby* de Bitcoin, sección 3.6).

Las capas están solapadas, no separadas en el tiempo, lo que significa que en un mismo bloque TUC pueden coexistir operaciones de varios rieles. Esto es posible porque e-Bitcoin puede manejar múltiples flujos de destino a la vez, diferenciándolos por la etiqueta del riel. Desde la perspectiva del registro TUC, quizás internamente marque las transacciones con su riel, pero siguen siendo parte de la lista única de transacciones del bloque. En términos de implementación, podríamos tener campos como [timestamp TUC, riel, detalles de transacción] para cada entrada. Así, el bloque TUC actúa como contenedor multi-capa.

La ventaja de esta arquitectura es que **no se necesita un calendario separado por cada red**. Todos utilizan el mismo, lo que retoma la idea de interpolaridad: diferentes redes (Bitcoin, Ethereum, etc.) tienen eventos injertados en la secuencia TUC. Además, e-Bitcoin puede de esta forma servir de **plataforma de pagos unificada**: un usuario corporativo podría agendar en la misma interfaz de calendario pagos a empleados en distintos países usando distintos criptoactivos (BTC, ETH, etc.), y TUC orquesta todo en la misma línea temporal, cambiando simplemente el riel para cada uno.

En la preparación del sistema, se define la **nomenclatura de rieles** y su significado. `PAY-` sugiere que son rieles de pagos. Podría haber otros tipos de rieles futuros (por ejemplo, `MSG-` para envío de mensajes certificados, o `DOC-` para notarización de documentos en otras redes), pero en el contexto dado los ejemplos se centran en pagos financieros. Cada riel se asocia a una red externa específica y a un conjunto de reglas (p.ej., *PAY-BTC* tendrá ciertas limitaciones de monto o formato de dirección propias de Bitcoin).

La existencia de capas solapadas no rompe la contabilidad lineal, pues todo queda asentado en e-Bitcoin; simplemente añade una dimensión de clasificación. Podemos imaginar que en el libro mayor interno de e-Bitcoin las cuentas se subdividen por rieles, pero el orden temporal sigue único. Para el usuario final, la nomenclatura `PAY-BTC`, `PAY-ETH` es explícita al crear una orden, garantizando que la intención de a dónde se dirige el valor quede clara. Y para la red, es la señal para enrutar apropiadamente más adelante (vía lobbies). En resumen, los rieles proporcionan **paralelismo lógico** dentro de la secuencia temporal unificada: permiten manejar múltiples destinos heterogéneos simultáneamente, manteniendo una sola secuencia de tiempo.

3.6 Lobbys por red: cada riel apunta a un lobby de una red externa

Para cada riel externo definido (como PAY-BTC, PAY-ETH, etc.), e-Bitcoin implementa un **Lobby** específico que actúa como puerta de enlace hacia la red correspondiente. Un *lobby de red externa* es esencialmente un módulo o componente donde se acumulan y preparan las transferencias salientes de e-Bitcoin antes de ingresarlas efectivamente en la otra blockchain. Puede visualizarse como la “sala de espera” de las transacciones para una red destino: una vez que una transferencia por riel PAY-BTC es procesada en e-Bitcoin (es

decir, se alcanza su tiempo y se sella en TUC), esta transferencia pasa al Lobby Bitcoin, desde donde se gestionará su materialización en la blockchain de Bitcoin real.

La relación riel-lobby es uno a uno: cada riel apunta exactamente al lobby de su red designada. Por ejemplo, el riel PAY-ETH tendrá un Lobby Ethereum; allí se reunirán todas las transacciones que e-Bitcoin debe enviar a Ethereum (como pagos a direcciones de Ethereum). Este lobby comprende tanto la lógica de traducción de la transacción al formato de la red externa (por ejemplo, construir una transacción Ethereum con el nonce, gas, etc., apropiados), como la interacción con esa red (quizá a través de un nodo conectado a Ethereum o mediante API si es un sistema permissioned).

En la fase de preparación, e-Bitcoin define estos lobbys: se configuran los parámetros de conexión y operación para cada blockchain externa soportada. Es posible que haya que proveer liquidez o fondos en esos lobbys (por ejemplo, un monedero de salida desde el que se pagarán los BTC, en caso de que e-Bitcoin custodie fondos puente). Alternativamente, el lobby puede funcionar solo como *instrucción* para que un tercero (como una federación de nodos) ejecute los pagos en la red externa. De cualquier modo, conceptualmente el Lobby es la **entidad encargada de realizar la transferencia externa cuando llega la orden desde TUC**.

Al tener lobbys diferenciados, e-Bitcoin puede tratar con las particularidades de cada red de forma modular. El lobby Bitcoin sabrá del tiempo de bloque de Bitcoin (~10 minutos), de sus comisiones, UTXOs, etc. El lobby Ethereum entenderá de gas, confirmaciones, smart contracts si se requieren, etc. Desde la perspectiva de TUC, estos detalles no importan durante la agenda o el procesamiento interno – simplemente se confía en que el lobby correspondiente tomará lo agendado y lo llevará a cabo. Así, la existencia de lobbys por red hace posible la interoperabilidad real: *TUC fija el cuándo, el lobby ejecuta el cómo en la red externa*.

Se puede ver al lobby como un análogo a un *mempool* o colador de transacciones para cada cadena externa. Una vez las transferencias entran al lobby, quedan bajo las reglas de esa red (por ejemplo, esperar ciertas confirmaciones). Sin embargo, e-Bitcoin siempre conserva una referencia local de lo enviado al lobby y su estado. En secciones posteriores se detalla la construcción de Merkle por riel (9.2) y la distribución (10), que son funciones conjuntas de e-Bitcoin y los lobbys.

En resumen, en la preparación del sistema se establece que *cada riel tiene su lobby*, y se configura la infraestructura para esos lobbys. Esto asegura que, cuando llegue el momento de una transferencia por un riel externo, haya un mecanismo ya preparado para **encaminar la transacción desde el mundo TUC al mundo de la blockchain externa** correspondiente. Los lobbys, por tanto, son componentes críticos para concretar la interpolaridad que TUC promete.

3.7 Puertas de entrada: definición de canales de recepción (base)

Antes de que existan solicitudes, el sistema también debe definir cómo ingresarán dichas solicitudes a la red TUC – es decir, las **puertas de entrada** o canales de recepción de nuevas órdenes de transferencia. En la configuración base, e-Bitcoin contará con al menos una puerta de entrada general a través de la cual todos los usuarios pueden someter sus

solicitudes (ya sean transacciones inmediatas o agendadas). Podemos imaginar esta puerta de entrada como la cola unificada de recepción, equivalente al mempool inicial donde llegan las transacciones pendientes antes de asignarse a un bloque.

La definición de estas puertas en la etapa preparatoria implica establecer la estructura de cola y las reglas básicas de admisión. Por ejemplo, se decide si habrá una única cola global para todas las transferencias entrantes, o si se segmenta por tipo (quizá diferenciando internas vs externas desde el inicio). La sección 6 hablará de escalar a múltiples puertas *solo si* hay saturación; por ahora, en la base, asumimos la configuración mínima: **una puerta de entrada unificada para toda la red**. Esta puerta aceptará solicitudes en tiempo real (que van para el bloque actual) y solicitudes agendadas (que van para bloques futuros), manejándolas inicialmente por orden de llegada y capacidad del sistema.

Configurar la puerta de entrada significa definir parámetros como: el tamaño máximo de datos que puede recibir por segundo, cómo se prioriza internamente (en principio no hay prioridades especiales, ver 5.5), y cómo se valida una solicitud antes de entrar (por ejemplo, verificar que esté firmada por una Key válida con fondos suficientes para cubrir la transferencia y posibles fees). Estas validaciones base aseguran que la puerta de entrada no se congestione con solicitudes inválidas o malformadas.

El concepto de **canales de recepción** también sugiere la posibilidad de tener distintos *endpoints* para recibir órdenes, quizás para diferentes perfiles de usuarios (corporativos, minoristas, APIs automatizadas, etc.). Sin embargo, de inicio, todos esos endpoints pueden simplemente alimentar la misma cola global. Lo importante es que el sistema sepa que toda solicitud entrante termina en algún *canal* donde esperará asignación a bloque.

En la práctica, la puerta de entrada base podría estar implementada en todos los nodos validadores: cada nodo mantiene en memoria las solicitudes pendientes (como un mempool distribuido). El consenso TUC (quizá conducido por el Block Creator o mediante un protocolo de consenso BFT) decidirá de ese pool qué entra en cada bloque. Para que eso funcione, deben definirse protocolos de gossip o difusión: cómo una solicitud enviada a un nodo se comparte con los demás para que todos la conozcan. Esto forma parte de la preparación: el **canal de recepción distribuido** debe estar listo para propagar rápidamente las nuevas órdenes a toda la red, garantizando que lleguen a tiempo al bloque correspondiente si aplica.

En síntesis, antes de que llegue la primera transferencia, e-Bitcoin ya tiene establecido por dónde entrarán esas transferencias. Las *puertas de entrada* son esa interfaz inicial entre los usuarios (o sus aplicaciones) y el motor temporal TUC. Con una configuración base consistente, la red está preparada para absorber la carga inicial de pedidos de manera ordenada, sirviendo de fundamento para luego, si fuera necesario, escalar a más puertas o canales especializados (como se verá en el escalamiento, sección 6). Pero todo parte de un diseño base sólido: **un canal de entrada común y transparente donde toda solicitud legítima puede entrar al flujo temporal consensuado**.

4. Planificación (proceso antes del tiempo presente)

4.1 Acceso al calendario TUC desde cada Key

En el sistema e-Bitcoin, cada usuario (identificado típicamente por una **Key** criptográfica, es decir, una clave pública/privada) tiene la capacidad de acceder al calendario TUC para consultar y programar eventos futuros. El acceso al calendario TUC se brinda mediante interfaces de usuario (por ejemplo, aplicaciones de monedero e-Bitcoin) y APIs que permiten explorar la **agenda global** de la red. Este acceso es universal y consistente: independientemente de la Key o el software utilizado, todos ven el mismo calendario consensuado.

Desde la perspectiva de experiencia de usuario, esto podría presentarse como un calendario interactivo donde se pueden seleccionar fechas y horas futuras disponibles. Por debajo, la Key (mediante su aplicación) está en realidad consultando la blockchain TUC (o un servicio asociado) para obtener información sobre los bloques. Es decir, la red ofrece funciones para *navegar* por el calendario, saber qué bloque corresponde a qué fecha y hora, y qué disponibilidad hay en términos de capacidad (si es relevante). Cada Key tiene permisos para leer el estado del calendario sin restricción, ya que es información pública del sistema. En este calendario unificado, un usuario puede, por ejemplo, desplazarse al mes siguiente y ver los bloques de cada día, incluso con detalle de si ya tienen transferencias programadas (no necesariamente mostrando detalles privados, pero quizás un indicador de ocupación/carga).

El acceso al calendario TUC es fundamental **antes del tiempo presente**, ya que es el paso inicial para planificar. Una analogía es cómo cualquier persona puede ver los horarios de un servicio de transporte para decidir en cuál quiere reservar; aquí las Keys ven los horarios de bloques. Al ser una red descentralizada, el acceso se implementa distribuido: cada nodo tiene la información del calendario (dado que está predefinido o por lo menos hasta cierto horizonte) y puede responder consultas. Es probable que los clientes mantengan una copia local de la plantilla de calendario para eficiencia, pero sincronizan con la red para asegurarse de estar actualizados en cuanto a asignaciones ya hechas.

En este contexto, “cada Key” implica que no se requieren privilegios especiales para interactuar con la agenda. Cualquier usuario con una Key válida puede intentar agendar un evento, sujeto a reglas de cuota o pago de tasa. No hay un intermediario central asignando turnos: es la interacción directa del usuario con la red la que realiza la reserva en el calendario. Esto empodera a los usuarios para **programar sus transferencias según sus propias necesidades temporales**. Por ejemplo, una empresa con su Key corporativa puede conectar al calendario TUC y pre-programar nóminas para los próximos 6 meses en los bloques correspondientes a fin de mes.

Por último, vale aclarar que el acceso al calendario no implica control unilateral: la Key puede proponer agendar algo en un bloque futuro, pero esa acción se convierte en una solicitud que debe ser admitida (y eventualmente confirmada) por la red. No obstante, el simple hecho de poder visualizar la línea temporal y seleccionar un bloque es ya un gran servicio proporcionado por e-Bitcoin, ya que da **transparencia temporal**. Cada Key sabe en qué momento se procesará su orden una vez agendada, lo que elimina incertidumbre. En síntesis, todas las Keys de e-Bitcoin tienen una ventana al futuro compartido de la red a través del calendario TUC, permitiéndoles planificar con la misma información y participar en la agenda común.

4.2 Selección de mes/día/bloque (agenda literal)

Una vez que un usuario (mediante su Key) accede al calendario TUC, el proceso de **planificación** consiste en la selección literal de la fecha y bloque deseados para su transferencia. Esto se asemeja a agendar una cita: se elige el **mes, el día y el momento específico** dentro de ese día. Dado que el calendario TUC está organizado por año/mes/día/hora, el usuario puede navegar por esta jerarquía hasta encontrar el bloque exacto que representa el minuto e incluso segundo que le interesa. Por ejemplo, si desea que su transacción ocurra el 15 de julio a las 09:30:00, buscará en la agenda del año correspondiente, dentro del mes de julio, el día 15 a la hora 9:30, y seleccionará ese bloque TUC.

Esta agenda literal implica que el sistema traduce la selección humana (fecha/hora familiar) al identificador de bloque TUC correspondiente. Muchas implementaciones probablemente harán esto automáticamente: el usuario escoge en formato calendario, y la aplicación determina "eso es el bloque #N". La importancia aquí es que TUC permite esa correspondencia directa, evitando cálculos complejos o incertidumbre – gracias a la plantilla predefinida, existe un mapping claro de fecha-hora a bloque.

Durante la selección, es posible que la interfaz muestre el **estado del bloque**: por ejemplo, si ya hay demasiadas transacciones planeadas en ese bloque, podría indicarse como "lleno" o "no disponible". O tal vez simplemente la red acepta sobrerreservas hasta cierto límite. Independientemente, el usuario literalmente está reservando un *espacio temporal* específico. Esta acción de seleccionar un bloque se convierte en parte de la orden que se enviará: la solicitud contendrá un campo indicando el bloque TUC de ejecución deseado.

En esta fase, el usuario debe también especificar los demás detalles de la transferencia (destinatario, monto, riel, etc.), pero aquí nos centramos en la **agenda literal**. El sistema e-Bitcoin actúa como un **calendario de reservas universal**: la selección de mes/día/bloque es análoga a reservar un recurso en un sistema compartido, solo que el recurso es un instante de procesamiento prioritario para esa transferencia.

Cabe resaltar que la agenda es global, así que mientras un usuario está seleccionando una fecha, otro usuario podría estar consultando la misma fecha. Si ambos intentan reservar el mismo bloque simultáneamente y no hay espacio para ambos, se aplicarán las reglas de admisión (por orden de llegada, presumiblemente). Desde el punto de vista del diseño, esto implica un mecanismo de sincronización: la solicitud final de agenda debe ser confirmada por la red para asegurar que esa reserva se concretó. Hasta no tener confirmación, la selección es tentativamente deseada pero no garantizada. En entornos de baja congestión, casi cualquier bloque futuro estará disponible; en entornos de alta demanda, los usuarios pueden tener que moverse en el calendario buscando un hueco.

La literalidad de la agenda en TUC es una gran mejora respecto a las blockchains tradicionales, donde uno no puede escoger el bloque futuro exacto en que su transacción caerá (solo se puede estimar). Aquí, en cambio, **la agenda es determinista y explícita**: el usuario señala un bloque futuro y, salvo que la cancele o falle por otra razón, su transacción ocurrirá en ese bloque específico. Esta capacidad de agendar literalmente permite casos de uso como: coordinar pagos múltiples de diferentes orígenes en el mismo segundo (pues

todos pueden apuntar al mismo bloque), o evitar días inhábiles (una empresa podría elegir que si cae en fin de semana, pase al lunes). En suma, la selección de mes/día/bloque le da al usuario control fino sobre el *cuándo*, convirtiendo la planificación de pagos en una actividad tan natural como poner un recordatorio en el calendario del teléfono, pero con la certeza de ejecución automática por la red.

4.3 Programación por ventana (10 min / 30 min / 1 hora)

Al planificar una transferencia, e-Bitcoin ofrece la opción de **programación por ventana de tiempo**, lo cual añade flexibilidad a la ejecución. En lugar de fijar un solo segundo exacto, un usuario puede especificar una ventana de tolerancia – por ejemplo, 10 minutos, 30 minutos o 1 hora – dentro de la cual su transferencia puede ser procesada. Esto significa que el usuario indica un bloque objetivo preferido, pero permite que la transacción sea ejecutada en cualquier bloque dentro de X minutos alrededor de ese objetivo, según convenga al sistema o a la optimización de capacidad.

La motivación de esta funcionalidad es gestionar de forma elegante posibles **picos de demanda o conflictos de agenda**. Supongamos que un usuario desea un pago alrededor del mediodía; en lugar de forzar exactamente 12:00:00, podría declarar una ventana de ±10 minutos. Así, la transferencia podría ocurrir a las 11:55, 12:00 o 12:05, según cómo esté de ocupado cada bloque, sin que ello afecte materialmente al propósito (quizá el destinatario no requiere la precisión al segundo, solo que ocurra cerca del mediodía). Para el sistema, esta flexibilidad le permite **reacomodar solicitudes en tiempo real** para evitar sobrecargas en un bloque muy concurrido: si 12:00 estuviese saturado pero 12:01 libre, el sistema podría correr automáticamente una transferencia con ventana de 10 min al bloque de las 12:01.

La programación por ventana se establece en el momento de agendar. El usuario elige un bloque deseado y opcionalmente selecciona una ventana (por ejemplo, en la UI podría haber un desplegable: “flexibilidad: exacto / ±10min / ±30min / ±1h”). Internamente, esto se traduce en permitir que la orden se incluya en cualquier bloque dentro del rango permitido. Probablemente, la solicitud enviada a la red incluirá el *rango de bloques admisibles* (por ejemplo, bloque 123450 ± 10 bloques). La red, al procesar, tratará de acomodarla lo más cercano al objetivo, pero tiene la libertad de correrla hacia adelante o atrás unos bloques si es necesario para manejar la capacidad.

Este enfoque mantiene la promesa temporal pero con margen: el usuario renuncia a un poco de certeza a cambio de mayor probabilidad de ejecución óptima. En escenarios de saturación, puede marcar la diferencia entre que su transacción entre puntualmente o se retrase más de lo deseado. Es importante señalar que si se especifica una ventana, el sistema no la usará salvo que sea necesario (no va a mover arbitrariamente las transacciones sin motivo). Solo en caso de congestión real del bloque objetivo, aplicará la ventana disponible. Esta característica también permite a los usuarios cooperar implícitamente con la red: usuarios que no necesiten extrema puntualidad pueden poner ventanas largas, dejando que la red aplane los picos de demanda.

Desde un punto de vista técnico, la implementación debe asegurar que **la transferencia no se ejecute fuera de la ventana**. Esto significa que si llega el límite superior de la ventana y aún no se procesó, en ese último bloque permitido *debe* entrar (de lo contrario, se estaría

violando la ventana prometida). Por tanto, la red tendría que priorizar absolutamente esas transacciones antes de que expire su ventana. Este es un compromiso: se gana flexibilidad pero se asume una responsabilidad de ejecución dentro del rango.

La disponibilidad de varias opciones (±10min, ±30min, ±1h) cubre diferentes casos de uso. Por ejemplo, un pago salarial podría tolerar ±1h sin inconveniente, mientras que un pago de arbitraje de mercado quizás necesite precisión exacta. Al ofrecer la elección, e-Bitcoin permite ajustar la calidad de servicio temporal a las necesidades del usuario. Incluso es posible que a mayor ventana se cobre menor fee, incentivando ventanas amplias cuando sea posible, pero eso es detalle de economía de red.

En resumen, la *programación por ventana* agrega resiliencia y eficiencia al sistema TUC. Los usuarios pueden dar cierto margen de maniobra temporal, facilitando que la agenda no se vuelva rígida ni frágil ante sobrecargas. Esto refleja un diseño inteligente de un calendario distribuido: mezcla la determinación firme con la adaptabilidad donde es aceptable, asegurando que **los eventos ocurran a tiempo, pero con espacio para optimización colectiva**.

4.4 Definición del riel en la orden: PAY-* y red blockchain destino

Al momento de programar una transferencia, el usuario debe **definir el riel** por el cual se ejecutará, lo que equivale a especificar la red blockchain de destino de dicha transferencia. Esta selección forma parte integral de la orden que la Key envía a e-Bitcoin. En la práctica, la interfaz de usuario presentará una opción para elegir el riel apropiado, comúnmente nombrado como `PAY-<blockchain>`. Por ejemplo, si el objetivo es enviar bitcoins on-chain, se escogerá el riel `PAY-BTC`; si son ethers en Ethereum, `PAY-ETH`; si la transferencia es puramente interna dentro de e-Bitcoin (sin salir a otra red), se podría indicar un riel especial o simplemente "interno" (quizás omitiendo `PAY-*` porque se sobreentiende).

Esta definición es fundamental porque determina **cómo y dónde la transferencia tendrá efecto final**. El riel identifica el *lobby* de salida a utilizar (como vimos en 3.6) y también indica a la red qué formato debe tener la información de la transacción (dirección destino, etc.). Por ejemplo, en una orden `PAY-BTC`, la dirección de destino provista deberá ser una dirección válida de Bitcoin. En una `PAY-ETH`, probablemente se especifique una dirección Ethereum y una cantidad en ETH (o tokens, según caso). La orden debe incluir todos los datos necesarios para que el lobby correspondiente construya la transacción externa llegado el momento.

Durante la planificación, esta selección de riel es tan importante como la elección del tiempo. De hecho, e-Bitcoin podría tratar como dos órdenes completamente distintas un pago a la misma hora pero con rieles diferentes. Esto es lo que permite la interoperabilidad: en un mismo bloque TUC, el usuario A puede tener un `PAY-BTC` saliente y el usuario B un `PAY-ETH`, y la red enrutará cada uno adecuadamente. Para el usuario, solo es cuestión de elegir el riel correcto. Cabe notar que *todos los rieles comparten la agenda*, es decir, no hay horarios separados por riel ni necesidad de coordinar relojes — la definición del riel simplemente encamina la transferencia dentro del bloque TUC asignado.

En la solicitud transmitida por la Key a la red, típicamente habrá un campo “network” o “rail” que indicará algo como: `rail: PAY-BTC`. Esta información va firmada junto con el resto de la transacción por la clave del usuario, de modo que no pueda ser alterada. Así, los nodos de e-Bitcoin al recibir la solicitud saben que esa transacción eventual deberá, tras procesarse, ser enviada al Lobby Bitcoin. Si el riel no coincide con el formato de datos (por ejemplo, se provee una dirección Ethereum con riel BTC), la solicitud sería rechazada en validación. Por tanto, la *definición del riel* también activa validaciones específicas de la red destino durante la admisión (4.5 o 5.1).

El soporte multi-riel permite a e-Bitcoin actuar como una **plataforma universal de pagos**. Pero también implica que los usuarios deben estar conscientes de cuál riel usar en cada caso: seleccionar el incorrecto resultaría en un fallo (un envío a una red equivocada). Para ayudar en esto, las aplicaciones pueden tener plantillas: p.ej., si uno escoge la moneda o el tipo de activo a transferir, la app automáticamente elige el riel adecuado. Incluso es posible un escenario donde un usuario transfiera un activo que existe en varias cadenas; entonces especificar el riel disipa ambigüedad (por ejemplo, USDT en Tron vs USDT en Ethereum, se podría reflejar con rieles distintos).

En suma, definir el riel en la orden es el paso en que el usuario le indica al sistema **qué “tren” tomarán sus fondos o datos al salir de e-Bitcoin**. TUC se encarga del *cuándo* (el bloque temporal) y gracias a los rieles determina el *dónde* (la red de destino). Con esta simple elección, el usuario integra en una sola programación lo que antes implicaría interactuar manualmente con diferentes sistemas. La orden quedará entonces completa: “Enviar X cantidad a Y destino en tal fecha mediante riel Z”. La claridad y explicitud de esta definición reduce errores y garantiza que cada transferencia programada esté correctamente *etiquetada para su ruta*, allanando el camino para su correcta ejecución en la fase de distribución.

4.5 Servicios típicos agendados: salario, alquiler, pensión alimenticia

El paradigma de TUC abre un amplio espectro de **servicios financieros y contractuales** que pueden beneficiarse de la planificación temporal. Entre los servicios típicamente agendados –aquellos pagos recurrentes o programados con anticipación– se encuentran:

- **Salarios**: Empresas y organizaciones pueden programar el pago de nóminas a sus empleados en fechas precisas, por ejemplo el último día hábil de cada mes a las 00:00 TUC. Gracias a e-Bitcoin, un departamento de RRHH podría cargar toda la planilla salarial una vez, indicando cuánto debe recibir cada empleado (vía su dirección, posiblemente interna o en otra red) y en qué bloque TUC (fecha) de cada mes se efectuará el pago. El sistema se encargaría de ejecutar estos pagos puntualmente. Esto garantiza que los salarios se paguen **en tiempo exacto** sin retrasos humanos, facilitando la planificación financiera tanto de la empresa como del empleado.

- **Alquileres**: Propietarios e inquilinos pueden acordar rentas periódicas (mensuales, trimestrales). El inquilino podría agendar en TUC transferencias `PAY-BTC` o internas hacia la dirección del propietario cada día 1 de mes a determinada hora. Alternativamente, un servicio de administración de propiedades podría agendar

pagos múltiples a diversos dueños en un mismo bloque (aprovechando la capacidad multi-transacción de un bloque). Con TUC, se elimina la excusa de “olvido” o tardanza: el alquiler se **programa una vez y ocurre automáticamente** en cada ciclo. Además, queda un registro temporal inmutable de cada pago de renta, útil legalmente para ambas partes.

- **Pensión alimenticia**: En casos de obligaciones legales como pensiones alimenticias o manutención, las cortes o las partes podrían utilizar e-Bitcoin para asegurar el cumplimiento puntual. Por ejemplo, una persona obligada a pagar una pensión podría agendar transferencias semanales o mensuales a favor del beneficiario (o de la cuenta judicial designada) con una ventana muy pequeña (casi exacta) para garantizar que llegue en fecha exacta según la orden judicial. Esto ofrecería tranquilidad al beneficiario y prueba de pago al pagador. Cada desembolso quedaría certificado en tiempo por TUC, sirviendo como evidencia en caso de disputas.

Estos son solo tres ejemplos, pero ilustran la potencia de TUC para **automatizar compromisos financieros periódicos**. El común denominador es que se trata de pagos que hoy en día suelen manejarse con sistemas centralizados (débitos automáticos bancarios, cheques posfechados, recordatorios manuales) y que a menudo sufren problemas de coordinación temporal (retrasos por fines de semana, errores de cálculo de fechas, etc.). Con TUC, todo esto se simplifica en *una instrucción programada una vez* que la red seguirá fielmente.

Otros servicios análogos podrían ser cuotas de préstamos, suscripciones a servicios, contribuciones a fondos de ahorro programadas, desembolsos de inversiones escalonados, etc. Incluso fuera del ámbito estrictamente financiero, se puede pensar en entrega programada de datos o documentos (por rieles especiales) – pero enfocándonos en finanzas, los salarios, alquileres y pensiones reflejan muy bien la utilidad concreta: garantizan que fondos cambien de manos **justo cuando deben hacerlo**, ni antes ni después, de manera transparente.

La agendabilidad de estos pagos mejora la **interoperabilidad comercial**: una empresa puede coordinar que sus desembolsos (salarios, proveedores) ocurran en sincronía con sus cobros (facturas de clientes) si todos usan TUC, manteniendo su flujo de caja con precisión de reloj. Además, la certificación temporal provee un historial incuestionable, lo que reduce disputas (“¿Cuándo me pagaste?” – “En el bloque TUC del 1 de marzo, como siempre, aquí está la prueba”).

En resumen, los servicios típicos agendados ejemplifican cómo TUC lleva la teoría a la práctica. **Salarios, alquileres, pensiones alimenticias** – pagos recurrentes que afectan la vida cotidiana – se vuelven eficientes, confiables y verificables mediante la agenda unificada de e-Bitcoin. Esto representa un avance significativo en la automatización financiera, alineando las transacciones con el calendario de la vida real de un modo jamás antes logrado con tal grado de certeza y descentralización.

4.6 Certificación agendada: HashPaper (SHA-256) + monto + riel

Además de transferir valor, la red e-Bitcoin puede servir para **certificar eventos o acuerdos ligados a una transacción programada**. Durante la planificación, es posible adjuntar a la orden de transferencia un concepto o documento hash, comúnmente denominado **HashPaper**, que proporciona un contexto o justificativo del pago agendado. Un HashPaper típicamente sería un hash SHA-256 de algún documento o conjunto de datos relevante (contrato, factura, acuerdo legal, etc.), el cual el usuario genera fuera de la cadena y luego incluye en la solicitud de programación.

La orden agendada así contendrá tres elementos esenciales: el **HashPaper (SHA-256)**, el **monto** a transferir, y el **riel** de destino. Por ejemplo, imaginemos un empleador programando el salario: podría adjuntar el hash de la nómina detallada o del contrato laboral del mes; monto: 500 eBTC (o equivalente en BTC por riel externo); riel: PAY-BTC hacia la dirección del empleado. Al hacerlo, la transferencia no solo moverá fondos sino que dejará anclado en la blockchain TUC un hash que representa la descripción de ese pago (por qué se hizo, bajo qué términos). Esto es extremadamente útil para **fines de auditoría y cumplimiento**, ya que permite correlacionar transacciones monetarias con documentos del mundo real sin revelarlos públicamente en la cadena (solo su huella digital).

El HashPaper funciona como **certificación agendada** en el sentido de que la red se compromete a registrar ese hash cuando procese la transferencia. Antes del tiempo presente, al confirmar la programación, e-Bitcoin valida que el hash esté correctamente formateado (256 bits) y simplemente lo almacena junto con la orden. Cuando el bloque TUC llegue, el hash se incluirá en los datos del bloque (por ejemplo, como parte de la transacción en una sección de memo/opreturn o en un Merkle aparte de documentos). Una vez minado/finalizado ese bloque, el HashPaper queda inmutablemente ligado a la transacción y a la marca temporal. Así, cualquiera con acceso al documento original puede recalcular el hash y compararlo con el almacenado en la cadena para verificar que coincide, demostrando que ese documento existía y era conocido al menos en el momento del pago.

Este mecanismo eleva el nivel de confianza: **la transferencia queda "sellada" no solo con hora sino con motivo**. Pensemos en una pensión alimenticia: el pagador podría incluir el hash de la sentencia judicial o del acuerdo de pensión. Cada cuota pagada va certificando que es conforme a ese documento base (si alguna vez se alega que la persona pagó menos de lo debido, se puede mostrar que el hash del acuerdo está referenciado en la cadena, vinculado a cada pago). En alquileres, se puede hashear el contrato de arrendamiento; en salarios, la nómina firmada por RRHH.

El concepto *HashPaper* sugiere la idea de un "papel" o documento digital cuyos detalles se condensan en un hash. SHA-256 es mencionado como algoritmo estándar por su robustez y porque es el mismo utilizado en Bitcoin y otras cadenas, lo cual facilita la interoperabilidad y familiaridad. No obstante, e-Bitcoin podría soportar múltiples tipos de hash; pero en su versión básica, SHA-256 es suficiente y universal.

Es importante notar que incluir un HashPaper no es obligatorio para cada transferencia; es una opción útil en contextos donde se necesita evidenciar el *qué* o el *porqué*. Si se deja vacío, la transacción sigue ejecutándose con solo monto y riel. Pero siempre que haya un compromiso formal detrás del pago, es aconsejable usarlo. La red trata el hash como un

dato inerte (no intenta interpretar su contenido), lo cual mantiene la privacidad del documento subyacente a menos que las partes voluntariamente lo revelen para verificación.

En resumen, la **certificación agendada mediante HashPaper** convierte a e-Bitcoin no solo en un scheduler de pagos sino en un *notario digital* vinculado a esos pagos. Combina la transferencia monetaria con la **prueba de integridad de documentación** asociada, proporcionando un sello integral: quién, cuándo, cuánto y bajo qué referencia se hizo cada transacción. Esto refuerza la confianza y seguridad jurídica de los pagos programados, uniendo el mundo financiero y el documental en un solo acto de programación.

4.7 Confirmación y acuerdo (si aplica)

En ciertos escenarios, la planificación de una transferencia puede requerir la **confirmación del receptor u otra contraparte** antes de volverse definitiva. TUC contempla la posibilidad de que algunas órdenes agendadas queden en estado *pendiente de acuerdo* hasta que la otra parte las valide, formalizando así un acuerdo bilateral (o multilateral) sobre la transferencia futura. Este mecanismo es útil cuando, por ejemplo, el receptor debe aprobar recibir fondos en una fecha determinada (quizá para confirmar la dirección destino o porque la transferencia cumple con ciertas condiciones acordadas).

4.7.1 Confirmación del receptor y “acuerdo sellado”

Si aplica la confirmación, el flujo sería el siguiente: el pagador agenda la transferencia en el calendario TUC con todos los detalles (monto, riel, fecha, HashPaper, etc.), pero en lugar de quedar inmediatamente confirmada, la orden queda marcada como “pendiente de confirmación del receptor”. El sistema notifica de alguna forma al receptor (por ejemplo, mediante su Key o un mensaje en su aplicación) de que hay una transferencia programada hacia él/ella a futuro, en espera de su aprobación. El receptor entonces revisa los detalles: verifica la fecha propuesta, el monto, y puede verificar el HashPaper (por ejemplo, calcular el hash del contrato que firmó para ver si coincide con el adjunto). Si todo es correcto, el receptor emite una confirmación digital (firma con su Key aceptando la orden).

Una vez el receptor confirma, la orden se convierte en un **“acuerdo sellado”**. Esto significa que ambas partes están de acuerdo y la transferencia queda firme en el calendario, con un estado irrevocable similar a cualquier transacción confirmada. La cadena TUC podría reflejar esta confirmación con, por ejemplo, un segundo hash o marca en la orden, o más sencillamente, incluyendo una transacción de confirmación vinculada. Lo importante es que a partir de ese punto, ninguna de las dos partes puede unilateralmente cancelar el pago sin incurrir en ruptura de contrato (y el sistema mismo no lo permitiría a menos que ambas lo acordaran de nuevo).

El concepto de “acuerdo sellado” alude a que la confirmación del receptor sella el compromiso – en esencia se convierte en una promesa de pago mutuamente acordada. Esto brinda mucha seguridad: por un lado, el pagador sabe que el receptor está enterado y de acuerdo en recibir los fondos en esa fecha (evitando problemas como que la cuenta de destino esté inactiva, o disputas sobre no haber estado al tanto). Por otro lado, el receptor tiene la certeza legal/moral de que el pagador *se comprometió*, con respaldo criptográfico, a ejecutar ese pago en la fecha indicada, lo que es más fuerte que una simple expectativa. En

ciertas jurisdicciones, esto podría incluso considerarse equivalente a tener un instrumento de pago diferido aceptado (similar a un cheque aceptado).

Desde la perspectiva de implementación, esta confirmación podría requerir que tanto el pagador como el receptor tengan Keys en e-Bitcoin (lo cual es usual si ambos son participantes). La orden original podría llevar una firma del pagador, y la confirmación añade la firma del receptor. Los mineros/nodos de e-Bitcoin verificarían que ambas firmas estén presentes antes de marcar la orden como confirmada definitiva.

4.7.2 Sin confirmación: permanece cancelable antes del presente

Si la confirmación *no* es aplicable o no ocurre (por diseño de la transacción o por falta de respuesta del receptor), entonces la transferencia agendada permanece **cancelable antes del umbral del presente**. Esto quiere decir que, hasta el momento en que se va a procesar (es decir, hasta justo antes de entrar en el bloque TUC programado), el pagador puede decidir cancelar la orden sin consecuencias en la cadena. También implica que, en ausencia de confirmación, la red no considera el acuerdo cerrado; es más bien una intención de pago abierta que el pagador podría retractar.

Esta política es importante para proteger al pagador en casos donde el receptor no confirmó o cuando surgen cambios. Por ejemplo, si alguien agenda un pago a un proveedor pero el proveedor no confirma estar de acuerdo con los términos (quizás porque hay un desacuerdo sobre el trabajo entregado), el pagador debería poder cancelar en tanto no se resolvió ese acuerdo. TUC permite esa cancelación siempre y cuando se haga **antes de cruzar el umbral del presente** para ese bloque (ver sección 7 sobre el umbral). En la práctica, el pagador emitiría una orden de cancelación (firmada) que la red procesaría inmediatamente, removiendo la transacción agendada de la cola futura. La cadena puede reflejar que la orden fue cancelada (posiblemente manteniendo un registro de que existió pero fue anulada).

El “sin confirmación permanece cancelable” también cubre casos de simple cambio de planes o error: si un usuario agenda algo por error en la fecha equivocada, puede cancelarlo oportunamente y quizás reprogramarlo correctamente. Desde luego, puede haber penalizaciones o condiciones para la cancelación (por ejemplo, podría perderse el fee ya pagado o parte de él), según políticas de red, para evitar abusos. Pero funcionalmente, la opción existe.

Es importante destacar que una vez se cruza el umbral del presente y el bloque se convierte en actual, la orden ya no es cancelable (se habrá ejecutado o estará ejecutándose). Por lo tanto, el pagador debe actuar antes de llegar ese momento. En la interfaz, seguramente habrá opciones claras de “Cancelar transferencia” disponibles hasta cierto límite de tiempo antes de la ejecución.

En resumen, la confirmación dual agrega una capa de confianza mutuamente reconocida: cuando se da, sella el acuerdo; cuando falta, mantiene la flexibilidad. No todas las transferencias requerirán esto – muchos pagos, especialmente a entidades pasivas o a uno mismo (ej: mover fondos entre propias cuentas a futuro), no necesitan confirmación de receptor. Pero allí donde haya dos partes con un acuerdo subyacente, esta funcionalidad asegura que TUC **no solo programa un pago, sino que puede representar un contrato**

aceptado. Y correlativamente, brinda la **válvula de escape** de cancelación al pagador si ese contrato no se cierra antes de tiempo, evitando ejecuciones no deseadas.

5. Ingreso de solicitudes (tiempo real o agenda)

5.1 Solicitudes agendadas: entran como órdenes para un bloque futuro

Las **solicitudes agendadas** son aquellas que los usuarios programan con anticipación para ser ejecutadas en algún bloque TUC futuro. Cuando una Key envía una solicitud de este tipo a la red, no se procesa inmediatamente en el bloque actual, sino que **entra al sistema como una orden pendiente** asignada al bloque objetivo especificado. En términos prácticos, la solicitud agendada es recibida por la puerta de entrada (mempool) de e-Bitcoin y marcada con el timestamp de destino. La red registra que “en el bloque TUC #X (fecha/hora Y) debe realizarse la transferencia Z”.

Estas solicitudes agendadas se almacenan en una especie de **cola futura**, organizada por orden temporal. Cada nodo de e-Bitcoin mantendrá una estructura de datos (p. ej., un calendario de transacciones futuras) donde coloca la solicitud en el “casillero” correspondiente a su bloque programado. Esta organización permite que, al llegar el momento (cuando el tiempo consensuado avanza al bloque X), el sistema sepa rápidamente qué transacciones estaban planificadas para incluir en ese bloque.

Importante destacar que la entrada de una solicitud agendada suele ocurrir con anticipación variable: puede ser minutos, horas o incluso meses antes del bloque de ejecución. Desde que entra al sistema hasta que se ejecute, esa orden permanece latente. Durante ese periodo, pueden suceder cosas: la orden podría ser confirmada por otra parte (4.7) si aplica, o el emisor podría cancelarla (antes del umbral, como vimos). Pero si nada cambia, la orden espera su turno. **Entrar como orden para futuro** implica también que deja de estar bajo el control activo del emisor (salvo cancelar) y pasa a ser parte del compromiso de la red: es decir, la red asume la responsabilidad de ejecutarla cuando corresponda.

Un detalle técnico: al ingresar una solicitud agendada, la red debería verificar que el bloque futuro referido efectivamente existe (según las plantillas de calendario) y aún no está saturado en capacidad. Si ese bloque no existiera (error de fecha) o ya tuviera tantas transacciones planeadas que excederían el límite, la solicitud podría ser rechazada inmediatamente o reasignada (quizá se pide al usuario elegir otro bloque o se autoajusta según ventanas de tiempo si las definió). Por lo tanto, en la admisión inicial de la solicitud agendada se hacen comprobaciones de viabilidad.

Una vez aceptada y guardada como orden futura, la solicitud agendada está sujeta al consenso de la red. Es decir, todos los nodos deben tener registro de ella para no olvidarla cuando llegue el momento. Esto se suele lograr difundiendo la solicitud a todos los nodos validadores, que la añaden a sus “agendas locales”. Dado que e-Bitcoin es una cadena con bloques cada segundo (por ejemplo), es posible que se confirme en la cadena principal una transacción que meta la orden en un bloque actual para consolidar la reserva. Sin embargo, una implementación más ligera puede simplemente acordar mempool futuro sin meter cada orden en la cadena hasta su ejecución. Los detalles variarán con el mecanismo de consenso.

En resumen, para cada solicitud agendada enviada, **la red la recibe y la pone "en fila" para el tiempo indicado**. No aparece inmediatamente en la cadena (puesto que su bloque aún no ha llegado), pero queda registrada en la memoria colectiva de la red. Cuando hablábamos de "agenda unificada", esto es justamente su reflejo: todos los nodos comparten una lista de tareas a futuro. La entrada de solicitudes agendadas, por tanto, es un proceso continuo de poblar esa agenda global con eventos por venir.

5.2 Solicitudes en tiempo real: entran al bloque del segundo actual

Además de las transacciones programadas con antelación, e-Bitcoin debe manejar **solicitudes en tiempo real**, es decir, aquellas que los usuarios desean ejecutar inmediatamente o lo antes posible. Estas solicitudes no llevan un bloque futuro específico, sino que al enviarse son candidatas a incluirse en el **bloque TUC correspondiente al segundo actual** (o al instante actual). En otras palabras, si un usuario pulsa "enviar ahora" en su aplicación, la solicitud llega a la puerta de entrada y la red intentará meterla en el bloque que se está formando en ese mismo momento.

El bloque del "segundo actual" es esencialmente el *próximo bloque a cerrar* en la cadena TUC. Dado el pulso constante, supongamos que estamos en el bloque con marca temporal 12:00:05; cualquier solicitud en tiempo real recibida durante ese segundo (y antes del cierre del bloque) puede ser incluida allí, dependiendo de la capacidad. Si llega después de finalizado, apuntará ya al siguiente (12:00:06). Por tanto, las solicitudes en tiempo real compiten para entrar en *el bloque de ahora* o a lo sumo en los inmediatamente siguientes si uno se llena o la orden llega tras el corte.

Para ilustrar: imaginemos a las 10:30:00.000 AM, inicia el bloque TUC correspondiente. Si a las 10:30:00.500 un usuario envía una transacción, esta es recogida por los nodos y, si hay espacio, se la inserta en el bloque 10:30:00 que se cerrará al completar el segundo. Si el sistema cierra bloques al final del segundo, la transacción entraría justo antes de 10:30:00.999. Una analogía es una puerta de embarque que cierra cada segundo, permitiendo entrar a quienes llegaron a tiempo.

En realidad, los detalles de sincronización fina pueden variar (posiblemente los nodos agrupan todas las recibidas en un tick de milisegundos y cierran el bloque). Pero conceptualmente, *solicitud en tiempo real = va al bloque actual*.

Este modelo se asemeja a cómo funcionan blockchains tradicionales: uno envía una transacción y espera que sea minada en alguno de los próximos bloques, idealmente cuanto antes. La diferencia aquí es la precisión temporal: en e-Bitcoin, "próximo bloque" significa *próximo segundo* (o intervalo definido), no un tiempo variable.

Las solicitudes en tiempo real son necesarias para escenarios no planificables o urgentes: por ejemplo, si ocurre un evento imprevisto y se necesita transferir fondos inmediatamente, o en trading de alta frecuencia donde se reacciona al mercado en ese instante. TUC no impide operaciones inmediatas; al contrario, las soporta integrándolas con las programadas. Ambos tipos de solicitudes conviven: unas esperando su futuro, otras compitiendo en el presente.

Un punto a destacar es que, a diferencia de las agendadas que ya tienen su “hueco” reservado, las en tiempo real **dependen de la disponibilidad** del bloque actual. Si el bloque actual tiene capacidad para, digamos, 100 transferencias y ya hay 80 planificadas en él (por ser, quizá, un momento top donde se agendaron muchos), entonces solo 20 nuevas en tiempo real podrían entrar. Si llegan 30, algunas quedarán fuera y deberán esperar al bloque siguiente. De esta forma, las solicitudes en tiempo real se ven afectadas por la carga ya existente (planificada o por concurrencia de otros realtime). Esto introduce la noción de que si uno requiere ejecución inmediata con total certeza, quizá conviene asignar su transacción a un bloque futuro cercano mediante programación (ej., unos segundos después) para garantizarlo; pero usualmente, en una red dimensionada adecuadamente, los bloques permitirán ciertos extras en tiempo real.

En conclusión, las solicitudes en tiempo real representan la **vertiente espontánea** de las transacciones en e-Bitcoin. Ingresan de inmediato al flujo y buscan cabida en el bloque presente. La red los maneja de forma análoga a un mempool convencional, con la salvedad de que el cierre de bloque es muy frecuente (cada segundo). Por tanto, la latencia entre envío y procesamiento puede ser de apenas segundos. Esto combina la ventaja de programabilidad (para lo anticipable) con la responsividad a eventos inmediatos (para lo inesperado), dándole a e-Bitcoin la versatilidad necesaria para un ecosistema financiero completo.

5.3 Sello de entrada: asignación al bloque TUC del segundo

Cuando una solicitud (ya sea agendada o en tiempo real) es recibida por la red e-Bitcoin, se le adjudica un **sello de entrada** que indica el bloque TUC en el que dicha solicitud ingresó al sistema. Este sello de entrada corresponde al tiempo presente en el momento de recepción – en el caso de una solicitud en tiempo real, será el bloque TUC actual (o inminente) en que se intenta procesar; en el caso de una solicitud agendada, podría ser también el bloque actual cuando se registró la orden como futura. En ambos casos, es un timestamp inmediato que marca la aceptación de la solicitud en la red.

El propósito del sello de entrada es proporcionar una **trazabilidad completa** del ciclo de vida de la transacción: no solo sabremos en qué bloque se ejecutó finalmente, sino también en qué bloque fue solicitada. Para solicitudes en tiempo real, estos dos sellos pueden coincidir (si se procesa en el mismo segundo que entró) o ser cercanos; para las agendadas, habrá típicamente una brecha significativa (la entrada pudo ocurrir hoy pero la ejecución en una fecha futura). Registrar el momento de entrada ofrece garantías de equidad (orden de llegada) y puede ser relevante en arbitrajes y auditorías. Por ejemplo, si dos solicitudes compiten por un lugar y solo una entra a tiempo, el sello de entrada demuestra cuál llegó primero.

Operativamente, en cuanto un nodo de e-Bitcoin recibe una nueva orden, podría encapsularla con metadatos como “[Recibida en bloque TUC #A (hora actual)] solicitando ejecución en bloque #B (si es agendada)”. Luego difunde esa información. El bloque #A es el sello de entrada. Si la orden es aceptada a la agenda, ese sello puede eventualmente quedar plasmado en la cadena quizás en forma de registro de admisión (dependerá si e-Bitcoin graba eventos de admisión en la cadena o solo los finales; podría que no los grabe en la cadena principal para no sobrecargar, manteniéndolo en logs distribuidos).

Desde un punto de vista de consenso, el sello de entrada es útil en condiciones de alta concurrencia. Supóngase un segundo saturado de peticiones en tiempo real: no todas podrán entrar en el bloque actual, algunas se deslizarán al siguiente. ¿En qué orden? Lo justo es que en el siguiente bloque se incluyan primero las que llegaron antes, en base a su sello de entrada. Así se evita la hambruna o que peticiones tardías se cuelen antes. En blockchains tradicionales, el orden de llegada es difícil de precisar globalmente por latencias, pero e-Bitcoin con bloques cada segundo puede usar la granularidad del segundo: las transacciones no incluidas en el bloque N tendrán sello de entrada = N, así que se les puede dar prioridad en N+1 sobre las que lleguen nuevas con sello N+1.

Por otro lado, en solicitudes agendadas, el sello de entrada también denota *cuánto antes fue programada* una transferencia. Esto puede tener relevancia, por ejemplo, para transparencia: se podría ver que un pago fue agendado con meses de anticipación (lo que puede implicar compromiso anticipado) versus uno agendado la noche anterior. Incluso podría ser importante en lotes de emisión: si hay una ventana de cancelación, se sabe cuánto tiempo estuvo la orden confirmada.

En síntesis, el **sello de entrada** es la marca temporal del nacimiento de la transacción dentro de e-Bitcoin. Complementa al sello de procesamiento (el bloque en que se ejecuta) brindando una visión de inicio a fin. A efectos prácticos, asegura el principio de *first come, first served* en la admisión y provee un rastro temporal adicional para fines analíticos y de confianza. Todo evento en e-Bitcoin queda así doblemente sellado: *cuando entró* y *cuando se procesó*, reforzando la integridad cronológica del sistema.

5.4 Admisión por capacidad: entra lo que permita la entrada de datos en ese segundo

La red e-Bitcoin tiene una capacidad finita por unidad de tiempo para procesar y almacenar transacciones, determinada por parámetros como el tamaño máximo de bloque por segundo o el número máximo de transferencias por bloque. Por tanto, la **admisión de solicitudes** en cada segundo está limitada a lo que la capacidad de entrada de datos de ese bloque permita. En términos simples, *en cada bloque TUC (cada segundo) solo puede entrar hasta cierto volumen de transacciones*. Todo lo que exceda ese volumen deberá esperar al siguiente segundo (o más allá).

Esta regla se aplica tanto a solicitudes en tiempo real intentando entrar al bloque actual, como a la cantidad total combinada de solicitudes programadas + en tiempo real en un bloque dado. Supongamos que la capacidad es de 100 transacciones por segundo. Si para el bloque de las 12:00:10 ya hay 80 transacciones agendadas, entonces en ese segundo solo 20 adicionales en tiempo real podrían ser admitidas; la 21ª tendrá que posponerse al bloque de las 12:00:11.

La "entrada de datos" puede referir también al tamaño en bytes. Quizá un bloque soporta, por ejemplo, 1 MB de datos. Si las transacciones son de tamaño variable (por adjuntar HashPaper u otras info), la admisión vendrá determinada por el llenado de ese espacio. En la implementación, los nodos al recibir nuevas transacciones chequean si al agregarlas al bloque actual se excedería el límite; si sí, entonces esas transacciones quedan en cola para

el siguiente. Esto es similar al funcionamiento de mempool tradicional con block size limit, pero en vez de esperar minutos, la espera es de un segundo a otro.

La **admisión por capacidad** asegura que la red no se sobrecargue más allá de lo que puede manejar en tiempo real, preservando estabilidad. También crea un comportamiento esperado en casos de saturación: si en un segundo llega una ráfaga masiva de transacciones (más de las que caben), se formará una fila que se irá despachando en los siguientes segundos. Esto implica que el tiempo de confirmación puede alargarse durante congestión, pero es fácil de cuantificar: si llegan 200 transacciones por segundo con capacidad 100, básicamente se generará un retraso de ~1 segundo extra, porque 100 entran ahora y 100 al siguiente segundo.

Notemos que las **solicitudes agendadas** ya reservadas cuentan contra la capacidad del segundo de ejecución. Es decir, la red al programarlas debe asegurarse de no sobrepasar la capacidad del bloque destino. Si inadvertidamente se agendaran 120 transacciones para un bloque de capacidad 100, la red tendría un dilema en ese momento. Por eso, en la fase de planificación se debe evitar tal sobresuscripción (posiblemente rechazando las extras o moviéndolas si hay ventana). La admisión por capacidad en tiempo real es la última salvaguarda: no entra más de lo que cabe.

Importante es que **no existe prioridad por "importancia"** (como se verá en 5.5). Así que la admisión es primero que llega primero que entra, hasta llenar espacio. En caso de congestión, transacciones equivalentes se diferirán sin favoritismo intrínseco, salvo quizás el orden de llegada o alguna política de fees (que no se menciona aquí, pero podría considerarse similar a una subasta de gas; sin embargo, dado "sin prioridad por importancia", se sugiere quizás no hay subasta de prioridad, simplemente orden temporal).

En conclusión, la admisión de solicitudes en cada segundo está regida por la **capacidad técnica del bloque de ese segundo**. Esto mantiene a e-Bitcoin operando dentro de límites seguros y previsibles. "Entra lo que permita la entrada de datos en ese segundo" encapsula esta idea: cada pulso temporal tiene un ancho de banda limitado, y el sistema se ajusta a él, distribuyendo cualquier excedente a pulsos posteriores mediante colas. El resultado es que e-Bitcoin puede absorber tráfico variable sin colapsar: en momentos de calma, la mayoría de segundos estarán subutilizados; en picos, se usarán a tope y formarán colas que se drenarán rápidamente a medida que los segundos pasan (o con medidas de escalamiento, como la sección 6 abordará).

5.5 Sin prioridad por "importancia"

Un principio explícito del sistema TUC es que **no existe prioridad basada en la "importancia" subjetiva de una transacción**. Esto significa que la red no discrimina las solicitudes por su tipo, monto, remitente/destinatario, o cualquier etiqueta semántica de importancia alegada. En la admisión y procesamiento, todas las transacciones son tratadas con igualdad de condiciones, siguiendo básicamente el orden temporal de llegada o de programación, y los límites de capacidad descritos, pero **sin colarse por consideraciones externas**.

En la práctica, esto implica que un pago pequeño de un usuario individual recibe el mismo trato que un pago multimillonario de una corporación, en términos de acceso a un bloque.

No hay un mecanismo intrínseco de “alta prioridad” que ciertos usuarios puedan invocar para saltar la cola (por ejemplo, no existe algo como marcar una transacción como “urgente” y que la red la adelante arbitrariamente). Tampoco hay distinción de rieles en cuanto a preferencia: un `PAY-BTC` no es más prioritario que un `PAY-ETH` o que una transferencia interna; todos compiten por espacio en igualdad dentro del bloque.

Esta filosofía de diseño promueve la **equidad y la neutralidad** de la red. E-Bitcoin actúa como infraestructura pública donde el derecho a ser procesado es igual para todos los participantes conforme a las reglas universales (tiempo, capacidad). Evita, por tanto, escenarios de censura o favoritismo a nivel de protocolo: ningún nodo debería reordenar transacciones dando preferencia a alguna por considerarla más importante que otra.

Cabe mencionar que en muchas blockchains la prioridad se suele establecer por fees (quien paga más fee obtiene prioridad). Aquí, la frase “sin prioridad por importancia” sugiere que no hay una priorización manual; es posible que e-Bitcoin use un esquema fijo de fees o cuotas que no afectan el orden (o incluso podría no haber fees diferenciados). Si bien el texto no menciona directamente fees, se infiere que la prioridad no se vende ni se concede: es puramente ordinal. Lo más cerca a prioridad que se permite es por orden de llegada (que no es prioridad en sí, sino justica temporal).

Esto tiene implicaciones: en momentos de saturación, algunas transacciones tendrán que esperar. Pero el sistema no juzgará “esta es más crítica que aquella” – simplemente atenderá en secuencia. Por ejemplo, si en un segundo saturado entró una transacción de una emergencia médica y otra de alguien jugando, la red no distinguirá: procesará las que llegaron primero o según cola. En cierto sentido, delega en los usuarios la responsabilidad de anticipar y programar con tiempo suficiente sus operaciones importantes para evitar choques. También podrían acordarse socialmente heurísticas (como usar ventanas de tiempo como vimos, para distribuir carga), pero no hay atajos de protocolo.

La ausencia de prioridad “por importancia” refuerza también la resistencia a manipulaciones: no se puede sobornar a la red con la etiqueta de urgente, solo con reglas formales (si existieran fees variables, sería oferta/demanda, no importancia intrínseca). Y en temas de interoperabilidad, garantiza que un riel no degrade a otro – todos los rieles son ciudadanos de primera clase en la cadena TUC.

En suma, e-Bitcoin implementa un sistema **imparcial** en la admisión y secuenciación de transacciones. Cada solicitud compite en igualdad de condiciones técnicas, sin consideraciones de quién la envía o por qué. Esto simplifica el protocolo (no hay capas de prioridad) y da garantías a los usuarios de que no habrá tratos preferenciales ocultos. Por supuesto, en caso de necesidades especiales, los usuarios cuentan con herramientas dentro del marco (p. ej., programar con anticipación, usar ventanas para aligerar congestión), pero no con privilegios. Este principio mantiene la ethos descentralizada: **la red sirve por igual a todos los participantes**, dejando que el consenso temporal sea el único árbitro del orden de ejecución.

6. Escalamiento de ingreso (solo si hay saturación real)

6.1 Condición estricta: demanda > capacidad de entrada del segundo

El mecanismo de **escalamiento de ingreso** en e-Bitcoin está diseñado para activarse únicamente bajo una condición estricta: cuando la demanda de transacciones entrantes supera consistentemente la capacidad de entrada de la red en cada segundo. En términos concretos, esto significa que se observa una saturación real y sostenida – es decir, en cada bloque TUC están quedando fuera más transacciones de las que pueden ser procesadas, resultando en colas crecientes. Solo al cumplirse esta condición (demanda > capacidad) se considera necesario habilitar medidas de escalamiento.

Esta filosofía es deliberada para evitar sobredimensionar o complicar la red innecesariamente. Mientras la carga de transacciones pueda ser manejada dentro de los parámetros normales (incluso si hay picos breves), no se introducirá complejidad extra. Pero si la saturación se vuelve una realidad palpable (por ejemplo, cada segundo llegan 150 transacciones y solo caben 100, generando retrasos acumulativos), entonces e-Bitcoin entraría en un modo de **escalamiento** para preservar el rendimiento y la usabilidad.

La “condición estricta” sugiere que debe haber evidencia objetiva de saturación. Esto probablemente se mide a través de métricas monitoreadas por los nodos por consenso: tasas de ocupación de bloques, longitud de colas, tiempos de espera medios. Quizá se define un umbral, por ejemplo: más del X% de los bloques en un periodo Y llenos al 100%, o backlog medio de Z transacciones por segundo durante W segundos consecutivos. Al cumplirse ese umbral, los nodos podrían acordar (vía protocolo o votación de gobernanza) que hay saturación real y que se habiliten las medidas de escalamiento.

Es clave notar la frase *solo si hay saturación real*. Esto implica que el escalamiento no es proactivo ni preventivo en tiempos de holgura; solo se activa como respuesta a un problema presente. Además, “real” sugiere que se descarta saturaciones artificiales o momentáneas insignificantes – debe ser algo persistente. Por ejemplo, un pico de 5 segundos no justificaría reorganizar la arquitectura; pero sí un estado donde constantemente hay más demanda que oferta de espacio.

En la práctica, antes de llegar a escalar, se espera que la propia dinámica de fees (si existiera) o la latencia disuada un poco la saturación – pero si la red es de propósito amplio con bajo costo, saturación global es posible (como a veces ocurre en blockchains populares). E-Bitcoin tiene planificadas medidas en ese caso (ver siguientes subpuntos).

Un ejemplo ilustrativo: imaginemos e-Bitcoin inicialmente dimensionada para 100 tx/seg. En sus primeros años, maneja 20 tx/seg en promedio, con picos de 80 – sin saturación real. Pero a medida que la adopción crece, empieza a promediar 90, con picos constantes en 100 saturando bloques. Si ocasionalmente hay un backlog de 5-10 tx que se resuelven en segundos, aún es manejable. Pero supongamos que la red llega a promedios de 120 tx/seg – cada segundo 20 quedan en cola, formándose colas de cientos. Ahí se declara saturación real. Entonces se procedería al escalamiento de ingreso.

En resumen, el escalamiento de las vías de entrada de transacciones es **condicional**: se aplicará únicamente cuando la red se vea realmente sobrepasada en su throughput base. Esto asegura que no se introducen complicaciones sino hasta que la necesidad lo amerita. La detección robusta de esta condición es vital para un correcto accionamiento de las

medidas, garantizando que la red e-Bitcoin se adapta de forma reactiva y eficiente a incrementos estructurales en la demanda sin sacrificar su estabilidad en tiempos normales.

6.2 Activación de puertas temporales (preventivo solo por saturación real)

Una vez detectada la saturación real, e-Bitcoin puede optar por la **activación de puertas temporales** como medida de escalamiento preventivo. Estas “puertas temporales” son canales de entrada adicionales que la red habilita para manejar el exceso de demanda, pero su activación es transitoria y condicionada a la saturación persistente. En esencia, son **vías de entrada paralelas** que se abren cuando la vía principal está congestionada, permitiendo repartir la carga y evitar cuellos de botella.

Llamarlas *temporales* enfatiza que no forman parte de la configuración permanente, sino que se habilitan y eventualmente podrían cerrarse una vez normalizada la situación. Esto sugiere un enfoque flexible: la red puede escalar horizontalmente abriendo más “puertas” de acceso de transacciones durante el periodo de alta demanda, y luego volver a operar con la puerta base cuando la demanda decae. Es un poco análogo a abrir carriles adicionales en una autopista solo en horas pico.

La activación es *preventiva* en el sentido de que, al ver saturación continua, la red toma esta acción para prevenir un deterioro mayor del rendimiento (por ejemplo, tiempos de espera crecientes). Pero la política indica: “solo por saturación real” – no se abrirán preventivamente sin motivo. La red confía en su capacidad base hasta que las métricas de saturación se cumplan, entonces reacciona.

¿Cómo podrían implementarse estas puertas temporales? Una posibilidad es a nivel de protocolo acordar que a partir del bloque TUC N, se habilitarán entradas segregadas: por ejemplo, se crea un segundo mempool o canal de ingreso. Quizá cada puerta temporal se asocia a un conjunto de usuarios (ver 6.3). Los nodos validadores empezarían a escuchar transacciones no solo en la puerta principal sino también en una dirección diferente para la puerta extra. Esto podría significar, por ejemplo, que se acepta un segundo bloque por segundo en paralelo, pero más bien lo veo como un pipeline paralelo de admisión.

Sin embargo, e-Bitcoin prefiere mantener un solo bloque TUC por segundo (pulso constante). Entonces, ¿cómo caben las puertas temporales? Una interpretación: no se generan más bloques, sino que se reorganiza cómo entran las transacciones en esos bloques. Quizá se designan “subcanales” y luego se intercalan transacciones de distintos subcanales en el bloque actual. O se reparten por milisegundos dentro del segundo. Una analogía de puertas en la entrada de un estadio: abres más puertas para que la gente entre más rápido, aunque al final todos entran al mismo evento (el bloque).

Posiblemente, en saturación, la red podría alargar temporalmente el pulso (no deseable) o – más probable – activar alguna forma de sub-bloques o microbloques que se consolidan. Pero no han mencionado alterar el pulso, así que me inclino a lo de subcanales.

Lo que resulta claro es la intención: **dividir la avalancha de entradas** en flujos manejables que no interfieran unos con otros. Por ejemplo, si una sola puerta se saturaba porque muchas empresas mandan pagos al mismo tiempo, abrir otra puerta y redistribuir ciertos

emisores a la nueva evitará que compitan directamente con los de la puerta original, reduciendo colas en cada una.

Debe haber consenso para activar y desactivar estas puertas. Quizá la red en sí (los nodos) las habilitan vía un soft-fork paramétrico o una señal. O tal vez e-Bitcoin viene preconfigurada con X posibles puertas extras que están normalmente cerradas y se abren con un trigger.

En cualquier caso, la apertura de puertas temporales es un **mecanismo de emergencia planificada**: la red se reconfigura dinámicamente para aceptar mayor throughput de entrada sin romper la unidad temporal. Y siendo temporal, se minimiza impacto cuando ya no es necesario (cierra esas vías para volver a la sencillez base, probablemente para ahorrar recursos o evitar fragmentación del mempool).

En resumen, e-Bitcoin puede aumentar su capacidad de admisión en momentos de saturación real mediante la **habilitación temporal de canales de entrada adicionales**. Esto es análogo a escalar linealmente la entrada de datos. Es preventivo en el sentido de contrarrestar proactivamente la congestión antes de que cause retrasos intolerables, y se limita a situaciones justificadas. Así, la red consigue adaptabilidad: en días de tráfico excepcional (por ejemplo, un evento mundial que cause muchas transacciones al mismo tiempo), abre más puertas; pasada la tormenta, vuelve a la normalidad. Todo ello manteniendo coherencia con el consenso (los bloques TUC siguen uno tras otro, solo que alimentados por varias colas en paralelo).

6.3 Puertas por consenso (por volumen estructural)

La decisión de cómo abrir esas puertas temporales no es arbitraria: e-Bitcoin recurre al consenso de la red y a criterios de **volumen estructural** para determinar la configuración de estas puertas adicionales. “Puertas por consenso” implica que la activación y disposición de esas entradas paralelas es acordada colectivamente por los nodos de la red, en lugar de ser impuesta centralmente. Se basa en datos de volumen de transacciones estructural – es decir, patrones constantes de demanda alta atribuibles a ciertos grupos de usuarios o ciertas características del tráfico.

La idea es segmentar el flujo de entrada de acuerdo al volumen que generan ciertas entidades o grupos. Si la saturación real proviene de un conjunto identificable de actores (por ejemplo, algunas empresas que disparan muchos pagos), tiene sentido crear puertas dedicadas o compartidas para ellos, aislando su impacto del resto de usuarios. Esto se haría “por consenso”, lo que sugiere algún proceso de negociación o acuerdo entre nodos sobre qué segmentos crear.

Quizá la red detecta que, de las 150 tx/s que llegan, 80 provienen de la Empresa X, 50 de un conjunto de startups YZ, y 20 del público general. Se podría entonces abrir una puerta separada para la Empresa X (dedicada) y otra compartida para las startups YZ, dejando la puerta original para el público general. Esto equilibraría las cargas.

A nivel técnico, “puertas por consenso” podría significar un par de cosas:

- Que la asignación de qué usuarios van por qué puerta está definida en el protocolo tras deliberación.
- O que la creación misma de una nueva puerta (nueva listening port, nueva mempool partition) se activa mediante señal de los nodos (un blockflag, etc.).

Volumen estructural implica que no es un reparto aleatorio, sino siguiendo la estructura del tráfico. Por ejemplo, si 30% del volumen es de micropagos IoT (muchos, pequeños, de diversos remitentes), tal vez se decide agruparlos en su puerta; si 50% es de un exchange grande, se le asigna su puerta.

Dividir por volumen mejora la eficiencia porque puede haber optimizaciones específicas: tal vez las puertas dedicadas pueden filtrar o procesar con parámetros adaptados al tipo de transacción.

En todo caso, es vital que esto se decida por consenso para evitar disputas y fairness. Por ejemplo, los nodos podrían votar o algorítmicamente acordar: "cualquier entidad que consistentemente use >X% del throughput, se le crea una puerta dedicada". O "agrupemos entidades medianas hasta que sumen ~Y% en una puerta compartida".

Veamos los subpuntos 6.3.1 y 6.3.2 que detallan:

- **6.3.1 Puertas compartidas: agrupación de empresas por volumen**: Esto sugiere clústeres de usuarios medianos en una puerta común.
- **6.3.2 Puertas dedicadas: empresa con volumen cercano o equivalente al umbral**: Sugiere una puerta exclusiva para un actor muy grande.

Así, las puertas por consenso se configuran atendiendo a la estructura:

- Se identifican aquellos generadores de alto tráfico.
- Por consenso se determina si merecen una puerta propia (dedicada) o se agrupan con otros en una puerta compartida.
- Luego se implementa esa división en la red.

Hay quizás consideraciones de seguridad y antimonopolio: dar una puerta dedicada a una empresa la aísla pero también le da garantía de throughput. Debe ser porque de lo contrario saturaría la red. Consenso se asegura de que esto no se use indebidamente (por ejemplo, una empresa podría querer una puerta propia para ventaja; pero solo la obtiene si realmente su volumen lo justifica, medido objetivamente).

En síntesis, **puertas por consenso** significa que la comunidad de nodos decide, basándose en análisis de tráfico, crear canales de entrada paralelos segmentados por grandes contribuyentes de volumen, ya sea de forma grupal o individual, para manejar saturaciones. Esta es una solución elegante: no censura ni excluye a los grandes usuarios, pero los canaliza separadamente para preservar la experiencia del colectivo. Todo ello se hace de forma transparente y acordada, garantizando que la estructura resultante de múltiples puertas refleje fielmente la distribución del uso de la red y que su introducción esté legitimada por la necesidad y acordada por la red en su conjunto.

6.3.1 Puertas compartidas: agrupación de empresas por volumen

En situaciones de saturación en las que varios actores medianos o múltiples entidades suman un volumen significativo, e-Bitcoin puede optar por **puertas de entrada compartidas** que agrupen a varias empresas (o usuarios) juntas, según sus patrones de uso. La idea es crear un canal de recepción dedicado a un conjunto de participantes que, individualmente, no alcanzan el umbral para una puerta exclusiva, pero colectivamente generan un alto tráfico. Al agruparlos en una misma puerta, sus transacciones ingresan por ese canal separado, descongestionando la puerta principal.

Por ejemplo, supongamos que hay 10 empresas fintech, cada una generando un flujo constante de transacciones considerable (digamos 5-10 tx/s cada una). Ninguna por sí sola satura el sistema, pero juntas suman 50 tx/s, lo cual es la mitad de la capacidad base. En periodos pico, su actividad combinada llena gran parte del bloque base, afectando a los demás usuarios. En respuesta, la red (por consenso) podría crear una **puerta compartida** para estas 10 empresas fintech. Todas ellas enviarían sus solicitudes a través de ese nuevo canal (quizá identificado como “Puerta-Grupo1” o similar). Su tráfico se procesaría junto en esa vía.

Esta puerta compartida actuará como un segundo embudo: dentro de ella, esas empresas aún compiten entre sí por la capacidad de la puerta, pero ya no compiten con las transacciones del público general o de otros grupos. Esto significa que su tráfico queda aislado: si entre ellas saturan su puerta, no dañarán al resto del sistema y viceversa. Y al dimensionar la puerta según su volumen agregado, se les puede dar suficiente ancho de banda.

La agrupación por volumen suele obedecer a **criterios estructurales**: quizás se agrupan por sector (ej. varias bolsas de valores cripto en una puerta, varios procesadores de pagos en otra) o simplemente por cuota de mercado (los top N usuarios medianos en volumen). El consenso definirá la agrupación ideal. Podría ser dinámico: si una empresa en la puerta compartida crece demasiado, tal vez deba migrar a puerta dedicada (punto 6.3.2).

Estas puertas compartidas aseguran equidad dentro del grupo (siguen sin prioridad individual, solo aislados del exterior). Además, facilitan optimizaciones: por ejemplo, se podría ajustar parámetros de latencia o fees en esa puerta según su perfil (quizá altas tasas pero mayor tolerancia a micropagos o picos concentrados al inicio de cada hora, etc.).

Desde el punto de vista de implementación, los nodos de e-Bitcoin podrían etiquetar transacciones en mempool con un ID de puerta (basado en la firma o dirección del remitente, preclasificándolos). Al armar bloques, el Block Creator o validador incluiría una cuota de transacciones de cada puerta. Esto requiere coordinar a nivel de consenso cómo mezclar transacciones de múltiples puertas en un solo bloque (posiblemente se reserva una porción de bloque para cada puerta). Alternativamente, cada puerta podría tener su propio mini-bloque que luego se combina Merkle en el bloque maestro, pero manteniendo un bloque TUC único. Son detalles posibles.

Lo importante es el concepto: **varias empresas comparten una puerta** para no saturar la principal. Un caso de la vida real similar es, en internet, asignar un canal dedicado a cierto tráfico (por ejemplo, dividir anchos de banda entre servicios). O en un supermercado abrir una caja para "varias compras grandes" separada de la fila general.

La meta es suavizar los picos y aislar comportamientos: si esas empresas coinciden enviando muchas transacciones a final de día, saturarán su puerta pero no frenarán al resto; si el público general tiene su pico al mediodía, no afectará a las empresas en su puerta.

En suma, **puertas compartidas** son un método de escalamiento en el cual e-Bitcoin agrupa a múltiples generadores de alto volumen en canales comunes. Esto mejora la gestión del tráfico al segmentarlo lógicamente, permitiendo que la red atienda a cada segmento con menor interferencia mutua. Por supuesto, sigue siendo la misma blockchain unificada: finalmente todas las transacciones acaban ordenadas en TUC, pero esta segmentación en la entrada las regula antes de mezclar, manteniendo la performance y la previsibilidad para todos los segmentos.

6.3.2 Puertas dedicadas: empresa con volumen cercano o equivalente al umbral

Si una sola empresa o entidad genera por sí misma un volumen de transacciones **muy alto, cercano al umbral de saturación** (o incluso equiparable a la capacidad base de la red), entonces la solución más apropiada de escalamiento es asignarle una **puerta de entrada dedicada** exclusivamente para su tráfico. En este caso, dicha entidad no compartiría el canal con ningún otro usuario; tendría una vía exclusiva para enviar sus solicitudes a la red e-Bitcoin.

Una empresa justifica una puerta dedicada cuando su nivel de actividad es tan grande que, de usar la puerta general, la saturaría casi completamente ella sola. Por ejemplo, imagínese un gigante de pagos que envía 80-90 transacciones por segundo de manera continua (sobre un límite de 100 tx/s). En la puerta principal, prácticamente acapararía la mayor parte del espacio, perjudicando a todos los demás. Separarla en una puerta propia permite que su inmenso flujo sea manejado aparte, y la puerta general queda libre para la multitud de usuarios más pequeños.

Tener una puerta dedicada implica que e-Bitcoin, por consenso, reconoce a esta entidad de volumen masivo y acuerda brindarle un canal independiente. Esta decisión no es para favorecer a la empresa per se, sino para proteger al sistema global de su impacto. De esta manera, la empresa enviará todas sus transacciones a través de su puerta, y esas transacciones no entrarán en la cola de la puerta general ni de puertas compartidas. Su tráfico se alimenta a los bloques TUC desde su canal propio, posiblemente con una porción del bloque reservada para él.

Desde la perspectiva de la empresa, una puerta dedicada garantiza que sus transacciones siempre tengan un camino disponible (salvo que sature incluso su propia puerta, cosa que si ocurre implicaría que su demanda excede la capacidad total asignada para ella). Para la red, es más fácil optimizar así: puede dimensionar la puerta dedicada al ritmo de esa empresa. Por ejemplo, quizás se le otorga hasta 50% del bloque, o algún arrangement de throughput garantizado.

Esto recuerda al concepto de *colocación preferente* en infraestructura, pero en vez de favoritismo, aquí se justifica por eficiencia y necesidad. Se debe también calibrar que la suma de lo dedicado más lo de otras puertas no exceda la capacidad total (o la extienda en escalamiento, según sea la estrategia).

Implementar una puerta dedicada podría significar que los nodos mantienen un mempool separado para esa entidad (identificada por sus direcciones origen), y al armar el bloque siempre toman X transacciones de ese mempool dedicadas, independientemente de lo de los demás. En el peor caso, si la empresa no usa todo su espacio, quizás se desperdicia algo de capacidad; pero si la saturación era constante, es preferible.

La gestión puede ser dinámica: supongamos la empresa dedicada baja su uso por algún motivo prolongado, la red podría en el futuro degradarla a una puerta compartida o cerrarle la puerta dedicada para reasignar recursos. Pero inicialmente, al detectarse volumen "cercano o equivalente al umbral", se le da su carril propio.

Este arreglo promueve una **convivencia armónica** de grandes y pequeños participantes: los grandes no aplastan a los pequeños porque se les confina en su carril; los pequeños no sufren retrasos insoportables; y la gran empresa tampoco sufre tanto de la competencia con miles de micropagos de usuarios, que podrían afectar la latencia de su operativa crítica.

Un ejemplo concreto: imagínese que un banco global implementa pagos masivos por e-Bitcoin para liquidaciones entre filiales y genera cientos de pagos por segundo. Sin una puerta dedicada, ese banco congestiona la red entera. Con la puerta dedicada, e-Bitcoin dice "ok, banco, usa esta entrada, te garantizamos tu throughput, pero no interfieras con el resto".

En conclusión, **puertas dedicadas** son la máxima medida de escalamiento a nivel entrada para casos extremos de volumen. E-Bitcoin las usa para encapsular el tráfico de una mega-entidad en su canal, preservando la funcionalidad de la red para todos los demás. Es una demostración de cómo la red se puede adaptar estructuralmente, reconociendo la realidad del uso: si un actor equivale a medio mundo en transacciones, se le da su puerta para que no colapse al "medio mundo" restante. Todo esto, insistiendo, decidido en consenso para mantener la transparencia y evitar privilegios injustificados – es privilegio solo en tanto se corresponde a su carga extraordinaria.

7. Cierre del pulso y procesamiento (umbral del presente)

7.1 Umbral del tiempo presente (concepto central)

El **umbral del tiempo presente** es un concepto central en TUC que delimita el instante exacto a partir del cual un bloque temporal pasa de futuro a presente, desencadenando el procesamiento efectivo de las transacciones agendadas para ese momento. En términos sencillos, es el **punto de no retorno** en la cronología: el momento en que el tiempo consensuado alcanza un bloque TUC específico y éste se convierte en el “ahora” de la red. Antes del umbral, todo lo asignado a ese bloque está en el futuro (y potencialmente modificable o cancelable); al cruzar el umbral, ese bloque se considera presente, se sella y sus transacciones se ejecutan.

Podemos imaginar el tiempo en e-Bitcoin como una serie de celdas (bloques) por las que avanza una ventana de presente. El “umbral del presente” es el borde delantero de esa

ventana. Se mueve continuamente, segundo a segundo (o al intervalo definido), abriendo paso al bloque siguiente y cerrando el anterior. En cada tick temporal, un nuevo bloque cruza el umbral para convertirse en presente, mientras los bloques anteriores pasan a ser pasado inmutable y los futuros siguen esperando.

Este concepto es crucial para la comprensión de *cuándo exactamente* se procesan las transacciones: ni antes ni después del umbral. Si una transacción estaba programada para las 12:00:00, el umbral del presente la alcanza en el momento que el reloj consensuado marca 12:00:00. Antes de esa marca, ninguna acción ocurre; al llegar la marca, se dispara el procesamiento. Se puede visualizar como una línea que recorre el calendario TUC de izquierda a derecha; a la izquierda de la línea, todo está procesado (pasado), en la línea misma se está procesando (presente), a la derecha está pendiente (futuro).

Lo que hace potente a este concepto es que define con precisión la **finalidad temporal** de los eventos. En sistemas sin un tiempo unificado, “presente” es ambiguo; en TUC, presente es un bloque bien definido. Además, el umbral establece una referencia para los usuarios: hasta que su bloque no llegue al umbral, saben que su transacción sigue en espera (y, si no confirmada, puede ser alterada); al cruzarlo, saben que la transacción se está ejecutando y pronto será historia registrada.

Desde una perspectiva técnica, los nodos de e-Bitcoin están de acuerdo en cuál es el bloque actual (presente). Esta sincronización es parte del consenso: cada nodo sabe la hora TUC actual con un pequeño margen de error de red. Posiblemente, la red utiliza un protocolo para moverse al unísono bloque a bloque (similar a cómo ranuras de tiempo son manejadas en sistemas tipo Cardano Ouroboros). El umbral podría definirse, por ejemplo: “un bloque se considera presente cuando más del X% de los nodos han emitido su sello de tiempo pasado ese punto” o, en implementaciones más sencillas, por un reloj de referencia integrando.

El “concepto central” subraya que todo el diseño de TUC orbita alrededor de este umbral. Muchas reglas se definen en función de él: cuándo expira la ventana de cancelación (justo antes del umbral, ver 7.4), cuándo se marca procesado (7.2), etc.

En resumen, el **umbral del tiempo presente** es la línea divisoria móvil en la temporalidad consensuada de e-Bitcoin que señala el comienzo del ahora para cada bloque. Constituye la piedra angular temporal: gracias a él, la red puede decir con exactitud “a partir de este instante, ya estamos en el bloque X, todo lo previsto para X se ejecuta, y ya no se aceptan cambios referentes a X”. Es lo que convierte al tiempo en TUC en una secuencia de compartimentos estancos: el umbral va cerrándolos uno tras otro, asegurando un orden y finalización claros a medida que el tiempo avanza.

7.2 Definición estricta de “procesado”: procesado = cuando cruza el umbral del presente

En el sistema TUC, se adopta una definición estricta y precisa de cuándo una transacción u orden se considera **procesada**: **únicamente en el momento en que cruza el umbral del tiempo presente**. En otras palabras, una transacción programada para el bloque TUC N se considera procesada exactamente cuando el bloque N se convierte en presente y sus

transacciones son efectivamente ejecutadas. Cualquier estado anterior (pendiente, confirmada-por-receptor, etc.) no cuenta como procesado; solo al atravesar el umbral y entrar en fase presente adquiere el estatus de procesado.

Esta definición elimina ambigüedades. Por ejemplo, en sistemas tradicionales uno podría decir “la transacción está procesada cuando se minó en un bloque” — aquí esa noción se afina: “minarse en un bloque” equivale a “pasó el umbral del presente para ese bloque”. Antes de eso, aunque estuviera confirmada por ambas partes o reservada en la agenda, no se le llama procesada. Podemos pensar que hasta ese instante, estaba en preparación o programada, pero no realizada.

¿Por qué es importante esta distinción? Porque establece claramente la **finalidad operativa** de las transacciones. Procesado significa que los saldos se han movido, los cambios de estado (como envio a lobby externo, apuntes internos) se han efectuado y la red reconoce ese hecho como parte de su contabilidad histórica. Antes de procesado, nada de eso ha ocurrido realmente en el ledger; después de procesado, es un hecho consumado e inalterable.

Por tanto, si un usuario pregunta “¿Se procesó ya mi pago del viernes a las 15:00?” la respuesta se basa en esta definición: si el bloque TUC de viernes 15:00 ha cruzado el umbral (es decir, ya es pasado/presente), entonces sí, está procesado; si ese momento aún es futuro, entonces no lo está.

La “definición estricta” también tiene implicaciones legales y contables: por ejemplo, para reconocer un ingreso se puede usar la marca de procesado. Antes de eso, aunque esté agendado, sería un derecho a futuro pero no un hecho contable. Muchas situaciones de negocio requerirán esa certeza.

Internamente, los nodos sabrán que procesar implica incluir la transacción en el bloque actual y ejecutar su lógica: debitar al origen, acreditar al destino (sea internamente o generando la transacción externa), registrar el HashPaper, etc., todo en ese momento. Hasta entonces, esas acciones no ocurren.

Cabe resaltar que esta definición alinea la terminología con la dinámica del sistema. TUC no consideraría una transacción confirmada-por-receptor como “procesada”; eso solo la hace “acordada”. Procesada es análogo a “liquidada” o “ejecutada”.

Dado este marco, la red y los usuarios comparten la misma expectativa clara: nada está verdaderamente hecho hasta el tick del reloj correspondiente. Esto también tranquiliza en cuanto a revertibilidad: antes del umbral, puede haber cancelación; después, ya es final.

En síntesis, e-Bitcoin define con rigor el momento de procesamiento de las transacciones: **el cruce del umbral temporal**. Esto proporciona un punto focal de sincronía y entendimiento: toda la red sabe que en ese punto específico el mundo cambia de estado respecto a la transacción. No antes, no después. Es la cristalización del evento en la historia inmutable de la cadena.

7.3 Irreversibilidad: el sello TUC y el bloque ya no cambian

Una vez que un bloque TUC cruza el umbral del presente y sus transacciones son procesadas, ese bloque se vuelve **irreversible**: su sello temporal y su contenido ya no pueden cambiar ni alterarse en lo absoluto. Esto refleja la propiedad de **inmutabilidad** típica de las blockchains, aplicada aquí con énfasis en la dimensión temporal. En e-Bitcoin, cuando un bloque se cierra como presente, queda sellado para siempre con su marca TUC y todos los datos (transacciones, hashes, etc.) que contiene.

La irreversibilidad significa, en primer lugar, que **no existen reorgs del tiempo** en TUC. A diferencia de Bitcoin, donde un bloque confirmado podría teóricamente ser sustituido por otro en una reorganización de cadena, en TUC el consenso del tiempo lineal evita esa clase de incertidumbre (porque presumiblemente e-Bitcoin usa un mecanismo de consenso que finaliza los bloques en tiempo real, quizás BFT, dado el pulso constante). Una vez pasado el segundo X: no hay vuelta atrás, el bloque X es historia y nada la va a rescribir.

En segundo lugar, implica que **el sello temporal** (la etiqueta del bloque) queda fijo. Esto es natural por definición, pero vale enfatizarlo: si un evento se registró en el bloque de las 12:00:00 del 1 de enero 2026, siempre permanecerá en esa posición temporal. No puede posteriormente migrarse a otro bloque, ni se puede insertar nada nuevo retroactivamente en ese bloque. Por tanto, el timeline es apéndice único e inamovible.

Esta irreversibilidad proporciona confianza total: una vez un pago o registro se procesó, las partes pueden actuar sabiendo que es definitivo. Por ejemplo, un beneficiario puede entregar un servicio tras ver la transferencia procesada porque sabe que no habrá sorpresas de rollback. Igualmente, un documento certificado con un hash en un bloque a cierta hora puede ser presentado legalmente, sabiendo que la cadena no va a reorganizar los eventos.

Además, la irreversibilidad simplifica la contabilidad interna: e-Bitcoin puede mantener un registro interno (ver 7.5) de en qué bloque se procesó cada transferencia y esa información nunca cambiará, facilitando auditorías.

Para los lobbys externos, esto también es vital: cuando e-Bitcoin pasa transacciones a un lobby de red externa, lo hace tras irreversibilidad interna. Es decir, no va a pasar transacciones que luego puedan ser revertidas en su libro primario. Esto garantiza consistencia con los sistemas externos (no va a suceder que e-Bitcoin mande un pago a la red Bitcoin y luego diga “oh, se revirtió internamente”).

Técnicamente, la irreversibilidad en e-Bitcoin puede provenir de su mecanismo de consenso. Si utiliza algo como un consenso bizantino con finalización inmediata, cada bloque TUC es inmutable desde que se firma/pasa. Alternativamente, si hubiera algún tipo de pow, se complicaría, pero dado que TUC asume un control preciso del tiempo, es probable use un consenso con finalización determinística.

En síntesis, **una vez un bloque TUC se vuelve presente y se procesa, queda congelado para siempre en la cadena**. Ni su timestamp, ni sus transacciones, ni su Merkle, nada puede cambiar. La flecha del tiempo de TUC es estricta y unidireccional; el pasado no se reescribe. Esto sienta las bases para la robustez del sistema: los usuarios confían en que el historial contable y temporal es inalterable, cumpliendo la promesa fundamental de una

blockchain de que lo confirmado es irreversible, pero aquí con la precisión temporal añadida.

7.4 Ventana de cancelación: solo antes del umbral

El sistema TUC otorga la posibilidad de **cancelar o abortar** una transacción agendada, pero establece con rigor que dicha cancelación solo puede ocurrir **antes de que el bloque correspondiente cruce el umbral del presente**. Esto define una *ventana de cancelación* clara: desde el momento en que se agenda (o confirma por receptor, si aplicaba) hasta el instante inmediatamente anterior a su procesamiento en tiempo real.

En la práctica, esto quiere decir que un usuario (generalmente el remitente) puede retractarse de una transferencia programada en cualquier momento **mientras el bloque temporal asignado siga en el futuro**. Una vez que el reloj consensuado alcanza ese bloque, la ventana se cierra bruscamente; a partir de ahí, ya no es posible abortar la operación, pues está en curso o completada. Esto es coherente con la irreversibilidad discutida: no se puede deshacer lo que ya entró al presente.

¿Cómo se implementa? Un modo es que la transacción agendada permanezca en estado “pendiente cancelable” hasta el último segundo anterior. El usuario podría emitir una instrucción de cancelación (firmada) que la red reconoce. Si la red recibe la orden de cancelación antes de procesar la transacción, simplemente no la incluye en el bloque al procesarla (y tal vez en su lugar deja un marcador de cancelación, o la elimina de la cola futura). La cancelación en este contexto es semejante a anular un cheque antes de que sea cobrado. Importante: la red debe sincronizar bien los tiempos para no permitir cancelaciones tardías (por ejemplo, qué pasa si un nodo recibe cancelación casi al mismo tiempo que ya estaba ejecutando? Debe definirse un margen de seguridad, e.g., no cancelar en el segundo inmediatamente antes, pero supongamos la red es lo bastante rápida).

Durante la ventana de cancelación, la transacción no es definitiva. Esto da flexibilidad a los usuarios para manejar cambios de planes o errores. Por ejemplo, si alguien programó un pago pero luego ocurre una disputa con el receptor, puede cancelarlo antes de la fecha programada. O si accidentalmente puso mal la cantidad, puede cancelar y reprogramar correctamente. Sin esta ventana, programar a futuro sería arriesgado (porque la decisión sería totalmente irrevocable desde el momento de agenda, lo cual no siempre es deseable).

Sin embargo, esta ventana tiene límites precisos. "Solo antes del umbral" significa que no se permite ninguna cancelación tardía durante el procesamiento. Una vez se entra en presente, incluso si por algún milagro técnico uno intentara mandar una cancelación concurrente, la red la ignorará. Es conceptualmente similar a deadlines: se puede cancelar hasta la deadline, después ya no.

El sistema podría incluso automatizarlo: cuando un bloque entra en presente, internamente "finaliza" todas las transacciones programadas para él, descartando cualquier señal de cancelación subsecuente. Y podría registrar si alguna fue cancelada justo a tiempo para no procesarla.

Es posible que haya penalizaciones o condiciones para cancelar (por ejemplo, no reembolsar un fee ya pagado, etc.), pero conceptualmente es libre mientras se haga a tiempo.

Esta ventana de cancelación también interactúa con confirmación del receptor: si el receptor ya confirmó, la cancelación tal vez no sea unilateral; podría requerir consentimiento (eso dependería de las condiciones del acuerdo). Pero en ausencia de confirmación (4.7.2), es completamente cancelable.

En resumen, **la ventana de cancelación le da al remitente la libertad de revertir una promesa de pago planificada, pero únicamente hasta antes de que esa promesa se cumpla**. Este marco temporal estricto asegura que hasta el último instante previo al presente, el control está en manos del usuario (o de las partes si hay acuerdo), pero justo cuando la hora llega, la red ejecuta con compromiso total sin permitir marcha atrás. Esta disciplina temporal protege tanto la flexibilidad del usuario como la fiabilidad de la ejecución: nadie puede frustrar un pago en el último milisegundo una vez que pasó el umbral, brindando certeza al beneficiario de que, llegado el momento, recibirá los fondos sin cancelaciones sorpresivas.

7.5 Registro contable interno: e-Bitcoin fija en qué bloque TUC se procesó

Una vez que una transferencia es procesada (es decir, cuando ha cruzado el umbral del presente y se ha ejecutado), la red e-Bitcoin mantiene un **registro contable interno** especificando exactamente en qué bloque TUC fue procesada cada transacción. Esto significa que para cada operación confirmada, e-Bitcoin anota el identificador temporal del bloque en que se hizo efectiva. En términos contables, es como asentar no solo la fecha sino la marca de tiempo precisa de la transacción en los libros mayores de la plataforma.

Este registro interno es esencial para múltiples propósitos: auditorías, conciliaciones, consultas históricas y referencias cruzadas con sistemas externos. Por ejemplo, si una empresa quiere verificar el pago de salarios de marzo, el registro interno le mostrará que la nómina se procesó en el bloque TUC 2026-03-31 18:00:00 (por decir algo). Esa información puede servir para generar reportes o para demostrar cumplimiento (imaginemos, ante un ente regulador, se puede proveer la evidencia de en qué momento exacto se realizó un pago determinado).

En la infraestructura de e-Bitcoin, este registro podría estar implementado simplemente como parte de la propia blockchain: dado que cada transacción reside en un bloque, su bloque de procesamiento ya es implícito. Pero la mención “registro contable interno” sugiere quizás que e-Bitcoin además podría llevarlo en estructuras adicionales indexadas por cuenta u orden. Por ejemplo, el sistema podría permitir consultar la historia de una cuenta: cada transacción listada con su bloque TUC de procesamiento. O podría haber un map de ID de orden agendada -> bloque de procesamiento final, útil para ver si y cuándo se cumplió la orden.

Fijar en qué bloque se procesó también es fundamental para la interoperabilidad con los lobbys externos (sección 10.3). Cuando e-Bitcoin distribuye a la red externa, hace referencia

a su propio registro primario. Un lobby externo al recibir, supongamos, un paquete de transacciones, puede saber: “estas provienen del bloque TUC #N”. Así, si necesita, puede también anotar eso, generando un lazo audit trail entre la cadena interna y la externa.

Desde la perspectiva del usuario, esto aporta transparencia total: cada transacción finalizada tiene estampado su "cuando exacto" dentro de e-Bitcoin. Esto se alinea con la filosofía de TUC de eliminar el doble tiempo: ya no es necesario tener tu reloj local, la plataforma te da la hora unificada de cada suceso financiero. En una contabilidad descentralizada, en vez de “valor fecha” o “fecha de valor” (como se usa en bancos), tenemos el “bloque TUC de valor”.

Además, este registro impide cualquier confusión en caso de litigios o discrepancias: si alguien dijera “no recibí pago”, se puede demostrar “la red lo procesó en bloque tal” (lo que se apoya con irreversibilidad; de ahí sabrás si fue a su dirección).

Cabe imaginar que internamente e-Bitcoin actualiza balances en el momento del bloque, y asocia esa actualización con el bloque ID. Por ejemplo, cada cuenta puede tener un ledger con entradas estilo: [bloque TUC X: -Y eBTC salida a Z], [bloque TUC X: +Y eBTC entrada de W]. Así, la contabilidad en la base de datos de estado de e-Bitcoin también está temporalmente etiquetada.

Resumiendo, e-Bitcoin no solo ejecuta las transacciones sino que **documenta internamente el contexto temporal de cada ejecución**. Esto convierte a la blockchain en un libro mayor donde el *cuando* está tan registrado como el *cuánto* y el *quién*. “Fijar en qué bloque se procesó” cierra el ciclo de vida de la transacción en TUC: desde la agenda (cuando se planeó) hasta la concreción (cuando se realizó), todo queda escrito. Y una vez escrito (irreversiblemente), se convierte en parte confiable de la historia económica registrada por e-Bitcoin.

8. Clasificación por destino (interno vs externo) tras el procesamiento

8.1 Transferencias internas (solo e-Bitcoin): no salen a redes externas

Una vez procesadas las transacciones en un bloque TUC, es necesario distinguir entre aquellas cuyo destino es **interno a e-Bitcoin** y las que van dirigidas hacia otras redes (vía rieles externos). Las **transferencias internas** son aquellas donde tanto el origen como el destino residen dentro de la plataforma e-Bitcoin, sin requerir interacción con ninguna blockchain externa. Tras su procesamiento, estas transferencias internas **no salen fuera** de la red; su ciclo de vida se completa enteramente en e-Bitcoin.

Ejemplos típicos de transferencias internas podrían ser:

- Movimientos de fondos entre dos cuentas de usuarios dentro de e-Bitcoin (quizá e-Bitcoin tiene su propio token o moneda de contabilidad).
- Liquidaciones entre sub-ledgers o "rieles" internos (si e-Bitcoin ofrece tokens propios, stablecoins, etc. circulando internamente).

- Pagos en los que el riel es “PAY-BTC” pero hacia una cuenta custodial interna (aunque en general PAY-BTC implicaría externo, digamos riel "INTERNAL" o default).
- Cualquier transacción de servicio interno, como ajustar balances, recompensas, etc., que no representen entrega a otro blockchain.

Cuando la red identifica que una transferencia es puramente interna, simplemente la registra en el ledger de e-Bitcoin en el bloque correspondiente y ahí concluye: los saldos de la cuenta de origen y destino se actualizan según la transacción, el HashPaper se almacena como prueba (si tenía), y la transacción se marca finalizada. No hay paso adicional de enviarla a un lobby externo porque no es necesario. Estos fondos no “saldrán” a Bitcoin, Ethereum u otra red, se quedan circulando dentro de la economía de e-Bitcoin.

Desde la perspectiva de e-Bitcoin, las transferencias internas son probablemente más eficientes de manejar ya que no implican crear transacciones en otras cadenas ni pagar fees externas. Pueden liquidarse al instante dentro de un bloque TUC y ya. Además, no conllevan riesgo de discrepancia externa (como que un pago a otra red pueda fallar por congestión en esa red, etc.). En cierto modo, e-Bitcoin funciona aquí como una contabilidad cerrada: los débitos y créditos internos se reflejan inmediatamente con final de bloque.

Para los usuarios, una transferencia interna puede presentarse como "moneda e-Bitcoin" movida de A a B. Por ejemplo, si e-Bitcoin es adoptado por bancos, dos bancos podrían mover saldos digitales a través de e-Bitcoin sin tocar la red Bitcoin en sí, beneficiándose de la rapidez y programabilidad de TUC.

En informes, estas transacciones internas figurarán en el log de e-Bitcoin con su bloque, pero no habrá un TXID en otra blockchain. Su única referencia es dentro de e-Bitcoin.

Es clave que la red sepa clasificarlas: un riel "PAY-BTC" indica externo, un riel "internal" o ausencia de "PAY-*" implicaría interno. Al procesar el bloque, e-Bitcoin revisa cada transacción: si es interna, la trata como final; si es externa, la prepara para la siguiente fase (sección 8.2).

Esta clasificación también evita enviar cosas innecesariamente fuera. Por ejemplo, si un usuario accidentalmente marcara una transacción como interna pero era para mandar BTC on-chain, no pasaría; de hecho, la riel lo define. Todo con rieles "PAY-*" son external a su respectiva red; lo demás, e-Bitcoin lo toma en casa.

En síntesis, **tras el procesamiento, las transferencias internas quedan resueltas completamente dentro de e-Bitcoin**. No requieren más pasos, solo se asientan en la contabilidad e-Bitcoin. Esto contrasta con las transferencias externas, que una vez procesadas internamente aún deben verterse a otra red para consumarse externamente, como veremos a continuación.

8.2 Transferencias externas (por riel PAY-*): se preparan para lobby de red externa

Por otro lado, las **transferencias externas** son aquellas cuya entrega final debe ocurrir en una blockchain o sistema externo, indicadas por su riel `PAY-*` correspondiente (como

`PAY-BTC`, `PAY-ETH`, etc.). Una vez que tales transferencias son procesadas en e-Bitcoin (es decir, el bloque TUC llegó y la transacción se ejecutó internamente, debitando al remitente y registrando la intención de pago externo), **no se consideran totalmente finalizadas** desde el punto de vista del usuario final: aún debe completarse la acción de enviar el valor o información a la red externa destino.

Tras el procesamiento en e-Bitcoin, estas transferencias externas **entran en fase de preparación para el lobby de la red externa correspondiente**. En términos prácticos, e-Bitcoin ahora debe preparar los datos necesarios para que el lobby (ver sección 3.6) pueda inyectar la transacción en la blockchain externa.

¿Qué implica esta preparación? Algunas tareas típicas:

- Agrupar o enlistar todas las transferencias que van hacia la misma red externa en un periodo dado (por ejemplo, todas las `PAY-BTC` procesadas en los últimos X segundos) para pasarlas juntas al lobby Bitcoin.
- Construir transacciones específicas del protocolo externo. Por ejemplo, para Bitcoin, crear transacciones Bitcoin con inputs (probablemente provenientes de una dirección controlada por e-Bitcoin, quizás un fondo consolidado) y outputs hacia las direcciones de destino indicadas en las transferencias. Para Ethereum, armar transacciones Ether o token desde una cuenta de e-Bitcoin a la cuenta destinataria.
- Calcular las estructuras Merkle (ver sección 9) si e-Bitcoin pre-agrupará varias transferencias en una sola transacción externa.
- Reservar fondos de liquidez si e-Bitcoin maneja fondos puente (por ejemplo, asegurarse que su wallet custodio en la red externa tiene saldo suficiente para cubrir los pagos).
- Marcar en el sistema interno el estado "enviado a lobby" pero a la espera de confirmación externa, si se lleva un tracking.

Durante este estado, la responsabilidad pasa gradualmente del dominio de e-Bitcoin al dominio de la red externa. E-Bitcoin actúa como un orquestador: sabe qué debe enviarse, instruye al lobby a hacerlo, y luego esperará confirmaciones de que en la red externa se concretó (aunque la contabilidad principal ya la hizo al procesar, la salida a la red externa es para cumplir la entrega real de valor).

Por ejemplo, consideremos un pago `PAY-BTC`: al procesarse en e-Bitcoin, supongamos e-Bitcoin debita la cuenta interna del usuario en eBTC (o su stablecoin etc.) y genera un registro "enviar X BTC a Y dirección". Ahora, en preparación, e-Bitcoin reúne esas órdenes, y el lobby Bitcoin, controlado por e-Bitcoin o afiliados, construirá una o varias transacciones Bitcoin por esos montos a los destinatarios. Previo a enviarlas a la mempool Bitcoin, puede optimizarlas (agrupar salidas, priorizar, etc.).

Durante esta fase, desde la perspectiva del destinatario final (por ejemplo, alguien recibiendo BTC on-chain), aún no ve los fondos en la red externa porque están "en camino" a través del lobby. Sin embargo, e-Bitcoin ya considera irreversiblemente que debe hacerlo (porque internamente la transacción está procesada).

La distinción es similar a un sistema bancario que debita tu cuenta y envía una transferencia a otro banco: tu cuenta ya está debitada (interno), pero hasta que el otro banco reciba (externo) hay una etapa intermedia.

E-Bitcoin llevará un control para asegurarse que cada transferencia externa procesada se entregue. Puede haber mecanismos de reintento si la red externa falla, etc., pero eso es parte de la fiabilidad (puede ser atendido en secciones 9.5.2 y 10).

Es clave que esto ocurra *después* del procesamiento TUC, no antes. E-Bitcoin no envía nada a la red externa hasta que la transacción no sea firme internamente, para no mandar pagos que luego se cancelen. Una vez firme, se envía.

En resumen, **las transferencias externas pasan a una fase post-procesamiento donde se preparan para su ejecución en la red destino**. La "preparación para lobby" implica consolidar y transformar las órdenes internas en transacciones externas reales. Una vez preparadas (y usualmente de inmediato), entrarán al flujo de distribución: el lobby tomará esas transacciones preparadas y las inyectará en la red externa, como se detallará en la siguiente fase (Merkle building y distribución). En este modelo, e-Bitcoin actúa como un *clearing house* que finaliza internamente y luego *settlement* hacia fuera.

9. Construcción de Merkle (posterior al umbral del presente)

9.1 Merkle interno de e-Bitcoin (para transferencias internas)

Después de procesar cada bloque TUC, e-Bitcoin lleva a cabo la **construcción de árboles de Merkle** para consolidar y verificar las transacciones ejecutadas. En primer lugar está el **Merkle interno de e-Bitcoin**, que abarca todas las transferencias internas de ese bloque (aquellas que, como vimos, no salen a redes externas). El objetivo de este árbol Merkle interno es proveer una única huella criptográfica que resume todas las transacciones internas del bloque, garantizando la integridad y permitiendo verificaciones eficientes de inclusión.

Dentro de cada bloque TUC, e-Bitcoin puede estructurar todas las transacciones internas como las hojas de un árbol de Merkle. Cada transacción (con sus detalles, como remitente, destinatario, monto, hashPaper si hay) se representa mediante un hash (por ejemplo, SHA-256) que constituye una hoja. Luego, las hojas se agrupan de a pares, se calculan los hashes padres, y así recursivamente hasta obtener la **raíz de Merkle interna** para ese bloque. Esta raíz resume todas las transacciones internas incluidas en el bloque de manera tamper-evident (cualquier alteración de una transacción cambiaría la raíz de Merkle de forma detectable).

Esa raíz Merkle interna puede almacenarse en el encabezado del bloque TUC, o en un campo especial. De esta forma, la cadena e-Bitcoin se hace más confiable: cualquier nodo o tercero puede validar que una transacción específica forma parte del bloque buscando su posición en el Merkle e incluso hacer pruebas de inclusión con las ramas correspondientes.

Esto es análogo a Bitcoin, donde cada bloque tiene un Merkle root de transacciones, permitiendo verificar transacciones sin descargar todo el bloque.

Para e-Bitcoin, el Merkle interno es importante no solo para la integridad del registro interno, sino también porque sienta la base para cualquier derivación de confianza. Por ejemplo, si un usuario quiere demostrar que un pago interno ocurrió, puede presentar la transacción y una prueba de Merkle que vincula esa transacción a la raíz del bloque (la cual a su vez está firmada/consensuada en la blockchain). Esto proporciona prueba de inclusión inmutable.

Además, mantener el Merkle interno es eficiente para sincronización de nodos ligeros: un nodo liviano de e-Bitcoin puede descargar sólo las cabeceras con raíces Merkle y luego solicitar pruebas para las transacciones relevantes, en lugar de todo el contenido.

En resumen, **tras procesar un bloque TUC, e-Bitcoin consolida todas sus transacciones internas en un árbol de Merkle**, obteniendo una raíz única que compendia el bloque. Este Merkle interno es fundamental para la integridad, verificación rápida y ahorro de espacio (almacenas una raíz en lugar de todos los detalles al referenciar un bloque). Es la garantía criptográfica de que el bloque no fue alterado, ya que la más mínima modificación en las transacciones cambiaría la raíz. Así, la contabilidad interna de e-Bitcoin se asegura eficazmente utilizando esta estructura probada de seguridad y eficiencia.

9.2 Merkle por red (por riel) construido en e-Bitcoin

Además del Merkle interno para transacciones dentro de e-Bitcoin, la plataforma también construye árboles de Merkle separados **por cada red externa (riel) hacia la cual se han dirigido transferencias** en un determinado intervalo. Esto significa que, tras procesar los bloques TUC, e-Bitcoin agrupa las transferencias externas de cada riel (como PAY-BTC, PAY-ETH, etc.) y para cada conjunto construye un **Merkle específico de esa red**. La finalidad es similar: crear un resumen criptográfico único que represente todas las transferencias destinadas a esa red externa en un periodo, sirviendo como un “paquete” o compromiso de esas transacciones.

¿Cómo funciona? Pensemos por ejemplo en riel PAY-BTC (transferencias hacia Bitcoin): E-Bitcoin, después de procesar X bloques TUC (o quizás a intervalos regulares, e.g., cada bloque TUC genera uno, o agrupan varios, eso lo veremos en 9.3), toma todas las transacciones de esos bloques destinadas a Bitcoin. Cada transacción externa (que tipicamente sería "Enviar M BTC a dirección D", con su hashPaper, etc.) se representa como una hoja (p. ej., su hash). Luego se construye un árbol Merkle con todas ellas, obteniendo una raíz de Merkle para el conjunto de transacciones `PAY-BTC`. Lo mismo se haría para `PAY-ETH`: tomar las transacciones a Ethereum, generar un Merkle y obtener su raíz.

El **Merkle por red construido en e-Bitcoin** tiene varias ventajas:

- **Compromiso cripto unificado**: Permite que e-Bitcoin cree un compromiso verificable de todo lo que planea enviar a la red externa, que puede incluso ser publicado o compartido. Por ejemplo, se podría publicar la raíz Merkle en un contrato

o en la blockchain externa para que los participantes externos tengan prueba de integridad del lote.
- **Agrupación**: Al tener todas las transacciones de una red en un árbol, e-Bitcoin puede tratarlas como un solo objeto (la raíz) para ciertos fines, como comunicarlo al lobby o, si la red externa lo soporta, entregar la raíz directamente.
- **Privacidad y eficiencia**: e-Bitcoin no necesita revelar los detalles de cada transacción si la red externa pudiera aceptar un Merkle: bastaría con dar la raíz y luego, bajo demanda, las ramas para probar transacciones individuales (esto es más teórico, en la práctica Bitcoin/Ethereum no aceptan merkle commitments de terceros a la ligera, pero veamos en 9.5).
- **Respaldo en caso de re-transmisión**: Si un lobby externo fallara y se vuelve a intentar incluir transacciones, la Merkle root sirve para verificar que las mismas transacciones se siguen considerando.

Es como crear un “bloque de transición” propio de e-Bitcoin para cada red externa, con sus transacciones como hojas y la raíz actuando como identificador de ese bloque de salida.

Estos Merkle por riel se construyen **en e-Bitcoin**, es decir, la responsabilidad recae en los nodos e-Bitcoin de agrupar y hash. Por ejemplo, después de cada 10 minutos, e-Bitcoin podría construir la Merkle de las transferencias Bitcoin de esos 10 min.

Este diseño sugiere que e-Bitcoin trata la salida a cada cadena externa casi como una sub-blockchain: con lotes (o micro-bloques) en forma de Merkle. Esto facilita la **extracción multi-bloque** (9.3).

9.3 Extracción multi-bloque: Merkle por red con transferencias de varios bloques TUC

Dado que las blockchains externas típicamente tienen cadencias de confirmación más lentas que el pulso de e-Bitcoin (por ejemplo, Bitcoin ~10 minutos por bloque), e-Bitcoin emplea un esquema de **extracción multi-bloque**: en lugar de enviar transferencias externamente bloque a bloque en tiempo real, agrupa las transferencias de varios bloques TUC en un solo árbol de Merkle por red. Es decir, el Merkle por riel que construye e-Bitcoin puede abarcar transacciones de múltiples bloques TUC consecutivos.

Por ejemplo, imagina que e-Bitcoin produce un bloque cada segundo. En 10 minutos tiene 600 bloques TUC. Sería ineficiente y excesivo intentar crear 600 transacciones separadas en la red Bitcoin para cada bloque, sobre todo si algunas tienen solo una o dos transacciones. En cambio, e-Bitcoin esperará a acumular cierto número de transacciones externas (o cierto intervalo de tiempo, como esos 10 minutos), y luego construirá un Merkle que contenga todas las transferencias a Bitcoin de ese periodo. Ese Merkle por riel representará las transacciones de *varios bloques TUC juntos*. A esta acción de consolidar transacciones de varios bloques en un solo Merkle la llamamos **extracción multi-bloque** (extracto en lote).

La idea de "extracción" sugiere que e-Bitcoin *extrae* las transferencias externas de sus bloques TUC, las agrupa, y genera un Merkle. De este modo, en lugar de tener un Merkle por red *por cada bloque TUC*, se tiene un Merkle por red *por cada conjunto de bloques TUC*

(ventana temporal). La ventana podría estar alineada con el tiempo de bloque de la red externa o con un número fijo de segundos.

Por ejemplo, puede definirse: "para Bitcoin, e-Bitcoin agrupa cada ~10 minutos de transfers en un Merkle y las envía en lote al lobby Bitcoin; para Ethereum, agrupa cada 1 minuto (porque Ethereum es más rápido), etc.".

Las ventajas de la extracción multi-bloque:

- **Eficiencia en la red externa**: Menos transacciones globales. En Bitcoin, en lugar de 600 micropagos, e-Bitcoin podría publicar 1 transacción con 600 outputs (o algunas transacciones) representando todas esas salidas, o un Merkle commitment optimizado.
- **Optimización de fees**: Unir muchos pagos en uno puede reducir la sobrecarga por transacción en la red externa (ej: aprovechar un solo input para múltiples outputs en Bitcoin).
- **Sincronización con ritmos externos**: Permite esperar al próximo bloque de la red externa para enviar los pagos de forma ordenada, en vez de inundar su mempool con flujos constantes.
- **Flexibilidad**: e-Bitcoin puede ajustar el tamaño del lote dinámicamente si la red externa está congestionada o no.

El Merkle por red multi-bloque aún permite anotar el orden temporal TUC de cada transacción (pues dentro del Merkle, cada hoja puede contener referencia a su sello TUC). Pero a efectos de la red externa, se ven como un lote.

También hay una cuestión de integridad: agrupar transacciones de varios bloques TUC en un Merkle significa que si se necesitaran evidencias, habría que quizá subdividir la prueba, pero la raíz de Merkle cubre todas.

El término "Extracción multi-bloque" sugiere una analogía: extrayendo (pulling out) transacciones de múltiples bloques TUC para armar el Merkle. Podría implicar que e-Bitcoin espera a tener un volumen suficiente de transacciones a una red antes de "extraerlas" a la red externa en un batch.

En resumen, **e-Bitcoin combina las transacciones externas de varios de sus bloques consecutivos en un solo paquete Merkle por red**. Esto equilibra la alta frecuencia interna con la menor frecuencia externa, manteniendo orden (pues dentro del Merkle se puede reflejar la secuencia TUC, pero se envía junto). Así, e-Bitcoin se adapta a la *cadencia/capacidad de la red externa* (como se detallará en 9.4) al empaquetar sus salidas en intervalos más amplios que un bloque TUC unitario.

9.4 Orden de inclusión: por sello TUC + cadencia/capacidad de la red externa (ej. Bitcoin ~10 min)

Cuando e-Bitcoin prepara las transacciones hacia redes externas, se esfuerza por conservar el **orden original según el sello TUC** de cada transferencia, al mismo tiempo que respeta la **cadencia y capacidad de la red externa** de destino. Esto quiere decir que, al construir los lotes (Merkle por riel) y decidir el orden en que las transacciones externas

serán incluidas en sus respectivas blockchains, e-Bitcoin las ordena primordialmente por su tiempo de procesamiento TUC, pero modulando la emisión para encajar en los intervalos de confirmación de la red externa.

Por ejemplo, consideremos Bitcoin donde los bloques son ~cada 10 minutos. Supongamos que e-Bitcoin tiene una serie de transferencias `PAY-BTC` procesadas en los últimos 10 minutos, con tiempos TUC específicos. Al formar el lote para Bitcoin, e-Bitcoin mantendrá la secuencia temporal: la más antigua (menor sello TUC) va primero en la transacción o el lote, luego la siguiente, etc., reflejando el orden en que se procesaron internamente. Sin embargo, e-Bitcoin solo enviará este lote al *momento adecuado*, digamos justo antes de un próximo bloque Bitcoin (o escalonadamente), considerando la cadencia de ~10 minutos.

Esto es importante: el orden temporal TUC es justo (quien estaba antes en TUC se intenta que esté antes en la transacción externa, o al menos en el mismo bloque externo si es un multi-output). Si hay tantas transacciones que deben repartirse en múltiples bloques externos, e-Bitcoin seguirá la cola temporal para asignarlas: las más viejas en el primer bloque externo disponible, las siguientes en el siguiente, y así.

La “capacidad de la red externa” también influye. Por ejemplo, Bitcoin tiene límite de tamaño por bloque. E-Bitcoin no debería intentar meter más outputs de los que caben en un bloque Bitcoin. Si tiene 1000 pagos de 0.001 BTC, quizás los divide en 2 transacciones con 500 outputs cada uno en dos bloques distintos, en lugar de una sola transacción gigante que podría exceder tamaño. O si Ethereum tiene límites de gas, e-Bitcoin agrupa de manera que no supere un cierto gas per block.

Además, e-Bitcoin puede adaptarse a la congestión: si la mempool de Bitcoin está muy llena y meter todo de golpe generaría delays, quizá e-Bitcoin regula su output (eso es sofisticado pero plausible con un lobby inteligente).

El ejemplo dado (Bitcoin ~10 min) ilustra que e-Bitcoin probablemente espera alrededor de 10 minutos de transacciones, ordenadas por su marca TUC, y las empaqueta para ser minadas en Bitcoin. Los pagos TUC que caigan dentro de esos 10 min, sin importar su sec, entran en el mismo lote. Si se extienden, van al siguiente.

Es como tener un convoy: los vagones (transacciones) se ordenan por hora de embarque, pero el convoy sale en intervalos fijos (los trenes de Bitcoin). Todos los pasajeros del último intervalo suben al tren en el orden que llegaron a la estación.

Mantener el orden por sello TUC es importante para fairness y para auditabilidad: uno puede correlacionar la secuencia interna con la externa. Por ej., si Bob fue procesado en e-Bitcoin antes que Alice, su pago idealmente aparece en Bitcoin no después del de Alice.

Claro que la red externa puede introducir variación (ej: un bloque se retrasa 12 min en vez de 10, o mineros incluyen transacciones en orden dif, aunque si e-Bitcoin hace un multi-output, el orden dentro de esa transacción se mantiene). En general, e-Bitcoin (a través del lobby) controlará este orden en la medida de lo posible.

Si un riel es Ethereum (~15s blocks), e-Bitcoin quizá lanza lotes más frecuentes, pero igual garantiza orden: un pago con sello TUC de 12:00:01 sale en un lote antes que uno de 12:00:30, etc.

En resumen, **e-Bitcoin ordena sus transacciones externas según su momento TUC y las lanza conforme a la ventana de tiempo y capacidad de la red externa**. Esto garantiza que la sincronía temporal interna se refleje lo más fielmente posible externamente, evitando alteraciones injustas en la secuencia. Al mismo tiempo, al adecuarse a la cadencia de la red externa (p. ej., 10 min), no sobrecarga esa red y aumenta la probabilidad de inclusión eficiente. Este equilibrio asegura que los pagos lleguen en la red externa en el orden esperado y dentro de un plazo razonable relativo a la expectativa temporal original de los usuarios.

9.5 Preconstrucción opcional de Merkle desde e-Bitcoin

En ciertos casos, e-Bitcoin puede optar por realizar una **preconstrucción opcional del Merkle** de transacciones externas desde su lado, con la intención de entregar a la red externa un paquete ya organizado y comprometido. Esto significa que e-Bitcoin no solo agrupa transacciones en un Merkle (como ya vimos), sino que incluso podría confeccionar completamente el bloque o porción de bloque de la red externa que las contendría. Sin embargo, esta estrategia depende fuertemente de si la red externa está dispuesta o habilitada para aceptar tal aportación foránea en su mecanismo de consenso.

9.5.1 Condición: depende de lo que la red externa permita aceptar

La posibilidad de preconstruir un Merkle para la red externa está condicionada a las reglas y flexibilidad de la blockchain destino. Algunas redes pueden permitir algo cercano a esto, por ejemplo:

- Redes con bloques propuestos por un consorcio o un contrato (donde se pudiera insertar un Merkle precomputado de transacciones).
- Sistemas con parachains o sidechains donde el aggregator (e-Bitcoin) podría inyectar su root directamente.
- O, conceptualmente, si un minero/pool de Bitcoin estuviera aliado con e-Bitcoin, podría incluir directamente un Merkle commitment. En la mayoría de redes clásicas (Bitcoin, Ethereum), la estructura del bloque es armada por sus propios mineros/validadores, no aceptando un "bloque listo" de afuera a menos que ese actor sea el validador mismo.

No obstante, hay tecnologías como **drivechains o merge-mining** donde un Merkle from sidechain is put in mainchain via special outputs (but that is beyond typical usage).

Así, e-Bitcoin podría, en el caso ideal, preparar la lista exacta de transacciones y su Merkle root, y solicitar a la red externa la inclusión de ese Merkle root en su próximo bloque. Esto solo pasaría si la red externa tiene un mechanism (ej. un smart contract or a soft fork arrangement) que acepta lotes de e-Bitcoin.

Por ejemplo, imaginemos una sidechain de Bitcoin piece: e-Bitcoin posts a hash of all outgoing transfers in an OP_RETURN on Bitcoin. Eso the network can accept (just data).

Sin embargo, para transferir actual value (BTC) a recipients, no basta un Merkle – se necesitan transacciones reales.

Quizás la idea es: si la red externa tuviera un contract/bridge that can decode a Merkle root posted by e-Bitcoin and then credit accounts accordingly. Por ejemplo, un smart contract en Ethereum where e-Bitcoin posts a Merkle root of all ETH transfers and a cryptographic proof that it locked corresponding tokens on e-Bitcoin side, then the contract itself releases tokens to addresses.

Eso suena a una suerte de Plasma or rollup approach.

Entonces, la “preconstrucción” se usaría si e-Bitcoin implementa un rollup on an external chain: e-Bitcoin could commit one hash to Ethereum representing many transfers, and maybe keep a liquidity pool to pay recipients via some mechanism.

Pero si la red externa no soporta esto (que es el caso de Bitcoin main: it doesn't parse Merkle from others spontaneously, except one could embed them as data only), entonces e-Bitcoin no puede forzar su block structure, y debe revertir a mandar transacciones tradicionales (lo que en la next bullet se sugiere).

En suma, e-Bitcoin evaluará: *¿la red externa me deja enviar un “batch cryptographic commit” en vez de transacción por transacción?* Si sí, aprovechará preconstruir Merkle; si no, no.

9.5.2 Si no lo permite: el lobby externo construye su propio Merkle

En la mayoría de casos con redes tradicionales (como Bitcoin, Ethereum), la red externa **no permite** a e-Bitcoin dictar la estructura del bloque: e-Bitcoin no tiene control sobre qué transacciones incluye un minero aparte de transmitírselas. Es el minero/validador de la red externa quien construye el Merkle de su bloque a partir de las transacciones en su mempool. Por tanto, si e-Bitcoin no puede inyectar un Merkle prehecho, entonces el proceso es:

- e-Bitcoin (a través de su lobby) envía las transacciones individuales o multi-output que representan sus transferencias, a la mempool o directamente a validadores de la red externa.
- La red externa toma esas transacciones como cualquier otras, las incluye en un bloque cuando corresponda.
- El **lobby externo** (que puede ser simplemente la red misma) construye su Merkle root normalmente incluyendo las transacciones de e-Bitcoin con las demás.
- E-Bitcoin no controla ese Merkle, pero puede verify after fact that its transactions got in.

Por "lobby externo construye su propio Merkle" entendemos: la blockchain externa seguirá su proceso usual de formar Merkle roots con todas las transacciones de su bloque (que ahora incluyen las de e-Bitcoin en la secuencia que el minero determinó).

Esto es más convencional: e-Bitcoin se comporta como un usuario masivo de la blockchain externa, sin cambiar sus reglas.

Un ejemplo: e-Bitcoin genera 1 transaction with 100 outputs to 100 recipients (a result of a Merkle grouping internally, but externally it's still 1 tx). Ese tx es enviado a Bitcoin mempool, un minero lo mete en un bloque junto con 2000 otras tx. El minero calcula el Merkle root de ese bloque (donde e-Bitcoin's tx is one leaf). E-Bitcoin no influyó en el root, salvo que su tx es part of it.

Entonces, si la red no lo permite (lo usual), e-Bitcoin no se beneficia de su precomputed Merkle, salvo usarlo internamente para track.

Y es probable que hoy, Bitcoin/Ethereum no permiten más. Ethereum has rollups concept, but bridging actual ETH require a contract where you deposit to release on other side, which is out-of-scope probably.

La frase "si no lo permite: el lobby externo construye su propio Merkle" sugiere resignación: e-Bitcoin envía transacciones al lobby (mempool) y los mineros validators manejan desde ahí.

En sumario, la **preconstrucción de Merkle** es una estrategia que e-Bitcoin usaría solo si la red destino tiene mecanismos para integrarla (tal vez en futuros, sidechain tech, etc.). En caso contrario, e-Bitcoin se atiene al método tradicional: envía transacciones, y la red externa los mete en su Merkle de la forma convencional sin intervención especial de e-Bitcoin.

Esto deja la puerta abierta a integraciones avanzadas: en un escenario ideal, e-Bitcoin y la red externa cooperan (tal vez e-Bitcoin validators are also validators on that chain, enabling trust). Sino, fallback.

En resumen, e-Bitcoin puede optimizar la entrega externa por precomputar un Merkle root de salidas y pedir que la red externa lo incluya (ej: via a collator or aggregator contract). Si la red externa no soporta tales atajos, e-Bitcoin envía las transacciones individuales y la red externa las incluye a su manera, construyendo su Merkle root normal. Esto reafirma que e-Bitcoin se adapta a la tecnología disponible de la red destino para garantizar que las transferencias se concreten de la manera más eficiente y segura posible, ya sea integrándose profundamente o usando el camino estándar.

10. Distribución hacia lobbys de redes externas (cuando aplique)

10.1 e-Bitcoin como distribuidor: enruta por riel hacia lobbys externos

Una vez e-Bitcoin ha preparado las transferencias externas (agrupadas en Merkle o transacciones listas según el caso), actúa como un **distribuidor** encaminando cada conjunto de transacciones hacia el *lobby* correspondiente de la red externa. En esencia, e-Bitcoin se convierte en un **router de pagos**: toma las transacciones de riel `PAY-BTC` y las envía al Lobby Bitcoin, toma las de `PAY-ETH` y las envía al Lobby Ethereum, y así sucesivamente para cada riel integrado.

El término “lobby” lo usamos para referirnos a esa instancia o mecanismo que interactúa directamente con la red externa (puede ser un nodo propio de e-Bitcoin en esa red, un servicio custodio, o un contrato puente). E-Bitcoin, al desempeñarse como distribuidor, ejecuta la lógica de entrega:

- Elige la secuencia y momento de envío de transacciones a cada red externa, de acuerdo con la planificación que hizo (por ej., justo después de minado un bloque Bitcoin previo, comienza a enviar las nuevas transacciones).
- Se asegura de dirigirlas al canal correcto: por ejemplo, se abre una conexión RPC o usa la API/nodo del Lobby Bitcoin para introducir las transacciones en la red Bitcoin.
- Monitorea que hayan sido recibidas correctamente (ej., verifica que aparecen en la mempool y luego en un bloque externo).
- Puede aplicar reglas específicas de cada red: por ejemplo, en Ethereum, definir gas price para las transacciones; en Bitcoin, ajustar fees (tal vez consolidó outputs, etc.).

E-Bitcoin asume un papel proactivo: no espera pasivamente que otro se encargue, sino que *ella misma* realiza el envío hacia las redes externas. Esto es importante ya que garantiza un control de extremo a extremo sobre la operación programada: la misma entidad (la red e-Bitcoin y sus nodos) que aceptó la orden y la procesó internamente, ahora la impulsa hasta su destino final.

Técnicamente, los nodos e-Bitcoin dedicados al output (quizá denominados *gateway nodes*) gestionan esta distribución. Podría ser que cada riel tiene uno o varios nodos interfase. Podría integrarse directamente con las API de los clientes de esas blockchains.

Por ejemplo, el Lobby Bitcoin podría ser simplemente una instancia del software bitcoind administrada por e-Bitcoin, a la que se alimentan las transacciones mediante sendrawtransaction. El Lobby Ethereum podría ser un nodo Ethereum o un smart contract.

Desde la perspectiva de un usuario, e-Bitcoin le ofrece un servicio integral: uno agenda un pago en e-Bitcoin y e-Bitcoin se encarga de “enrutarlo” a la red destino, cumpliendo la función que manualmente haría un pagador (firmar y emitir transacción en la otra blockchain).

Se usa la palabra "distribuidor" porque e-Bitcoin reparte a cada red lo que le corresponde: la analogía es una central de correos que clasifica cartas por país y las envía a los servicios postales de esos países.

Esto es “cuando aplica” porque no todas transacciones requieren esto (solo las externas).

En suma, e-Bitcoin como distribuidor garantiza que el trabajo iniciado en su plataforma alcance efectivamente su conclusión en la plataforma destino. **Enruta por riel hacia los lobbys externos**, significando que con base en la etiqueta riel, sabe exactamente a qué “puerta de enlace” dirigir cada flujo. Este routing es crítico para lograr la interoperabilidad: sin él, e-Bitcoin sería una mera agenda, pero con él, se convierte en un sistema de pagos/entregas multi-red cohesionado.

10.2 Cada lobby externo procesa su flujo según sus reglas

Una vez que e-Bitcoin entrega las transacciones al lobby de la red externa correspondiente, **cada lobby externo procesa el flujo de transacciones de acuerdo a las reglas y mecanismos propios de su red**. Es decir, después de la distribución, la responsabilidad primaria pasa a la red externa: esas transacciones entran en el ciclo normal de confirmación de Bitcoin, Ethereum, u otra cadena, y se someten a las normas de dicha red (tiempos de bloque, consenso, fees, ordenamiento en mempool, etc.).

Por ejemplo:

- En el lobby Bitcoin: las transacciones enviadas por e-Bitcoin entran al mempool de Bitcoin. Luego, los mineros de Bitcoin las evaluarán según su política (típicamente por fee rate) y eventualmente las incluirán en un bloque. Una vez incluidas en un bloque Bitcoin y confirmadas con la cantidad deseada de confirmaciones, se considerará completado el pago en Bitcoin. En este trayecto, e-Bitcoin ya no tiene control exacto sobre cuál minero la incluye ni en qué bloque exacto (salvo lo influya via fee).
- En el lobby Ethereum: las transacciones entrarán a la mempool Ethereum, competirán por gas price. Los validadores de Ethereum las incorporarán en bloques (posiblemente rápido dado su 15s time). E-Bitcoin debe asegurar poner un gas price razonable para su inclusión. Tras unos bloques, la transferencia se finaliza en Ethereum.
- Si es un lobby de una red con validación final (p.ej. un consorcio), quizá el flujo es directo pero igual se siguen sus protocolos (por ejemplo, waiting for finality).
- Si es un Smart Contract Bridge, el lobby puede implicar procesar un Merkle commit on chain, etc., siguiendo contract logic.

"Según sus reglas" enfatiza que, una vez en la red externa, e-Bitcoin no puede imponer su temporalidad ni prioridades. Tiene que jugar con las reglas:

- Prioridades por fees: e-Bitcoin debió asignar fees/gas adecuados. Si sus transacciones pagan muy poco, la red externa las puede demorar.
- Riesgos de reorg: en Bitcoin, e-Bitcoin puede esperar varios bloques de confirmación antes de fully concluding the process from its perspective.
- Finalidad condicional: Ethereum pos-PoS tiene finality after ~2 epochs (~13 min). E-Bitcoin puede decidir esperar para asegurarse.

Así, e-Bitcoin debe monitorizar cada lobby:

- Comprueba que las transacciones enviadas efectivamente se confirmen en la red externa.
- Maneja excepciones: si por alguna razón una transacción externa no se confirma (por ex, fee demasiado bajo, la red congest, double spend attempt improbable, etc.), el lobby puede reintentar con fee mayor u otro plan.

Pero críticamente, e-Bitcoin no modifica la semántica:

- Un pago `PAY-BTC` se convierte en una transacción Bitcoin usual. E-Bitcoin no puede, por ejemplo, confirmar en su interior más rápido de lo que Bitcoin lo hace. Debe esperar la confirmación real de Bitcoin para considerar que la entrega

completó. Mientras, internamente ya dedujo fondos, por lo que confía en que la red externa honrará. Este es un punto: hay un potencial riesgo que e-Bitcoin ha de asumir/per misrare (p.ej., se supone e-Bitcoin tiene los BTC depositados para enviar, so recipients will get them, so likely e-Bitcoin locks those coins long before).

Cada lobby opera su flujo: un lobby quizás representa un full node, que al ver sus transacciones lanza su propio mini-proceso (ex: un nodo Bitcoin difunde la tx a peers, un pool la mina, etc.). E-Bitcoin confía en los mecanismos descentralizados de la red externa para completar la tarea.

Se puede decir que *el registro primario queda en e-Bitcoin* (como en 10.3), pero la *ejecución finalizable monetaria* se consuma en la red externa.

Entonces, a grandes rasgos: **Cada lobby externo** actúa como traductor: toma el input de e-Bitcoin (transacciones a enviar) y lo deja en manos de su red, donde se procesa de la manera habitual para esa red. E-Bitcoin supervisará pero no interfiere con el consenso de la otra red.

Por lo tanto, para el usuario receptor: Si recibe en Bitcoin, verá quizás la transacción de e-Bitcoin en un explorador Bitcoin con sus confirmaciones, sin necesitar saber nada de TUC; para él es un pago normal de Bitcoin (quizás proveniente de una dirección de e-Bitcoin's custody). Si en Ethereum, similar, verán un tx del address de e-Bitcoin deposit.

La experiencia desde e-Bitcoin es fluida: "programaste un pago, e-Bitcoin lo procesó, y se mandó a la red externa, ya solo es esperar confirmaciones".

En resumen, **una vez e-Bitcoin entrega las transacciones al lobby externo, cada red las procesa de acuerdo a su protocolo**: e-Bitcoin delega la fase final al ecosistema natural de la cadena destino. Este acoplamiento flexible es crucial: no intenta imponer su propio consenso en otras redes, sino que integra con ellas, permitiendo que los pagos fluyan de TUC al mundo descentralizado más amplio sin fricciones más allá de las inherentes a cada blockchain.

10.3 Referencia permanente: el registro primario queda en e-Bitcoin por bloque TUC

Aunque los pagos y transacciones se terminan ejecutando en las redes externas, e-Bitcoin mantiene una **referencia permanente** de todos esos eventos en su propio registro primario, asociado al bloque TUC correspondiente en que fueron procesados. En esencia, la blockchain e-Bitcoin funciona como el libro de actas maestro donde cada transacción, ya sea interna o externa, quedó anotada en determinado bloque TUC, con su hash identificador y detalles. Esa anotación permanece para siempre como la evidencia de que e-Bitcoin procesó esa transferencia.

Esto tiene varias implicaciones:

- **Auditabilidad consolidada**: Si más tarde se quiere revisar qué pagos se hicieron, no es necesario ir a buscar en múltiples blockchains externas: uno puede consultar la blockchain e-Bitcoin como fuente primaria. Por ejemplo, un regulador podría

auditar e-Bitcoin y ver "en el bloque TUC 10500 se programó y ejecutó un pago de X BTC al address Y". Luego con esa info puede cruzar en la blockchain Bitcoin que efectivamente en la transacción tal se movió ese monto a Y, pero la garantía de la existencia del evento ya está en e-Bitcoin.

- **Resolución de disputas**: Supongamos un destinatario alega "no recibí fondos". E-Bitcoin puede mostrar en su ledger que la transacción fue procesada en bloque TUC N y enviada externamente. Si la red externa la confirmara, genial; si algo falló, esa discrepancia se detectará (y e-Bitcoin tendría que resolverlo, quizás reintentando o devolviendo fondos). Pero la referencia en e-Bitcoin permite rastrear el intento y su estatus.
- **Punto de sincronización**: La referencia en e-Bitcoin actúa como ancla temporal para coordinar con la red externa. Imaginemos que, por alguna razón, un lote no logró confirmarse en la red externa (tal vez un fee muy bajo y se quedó fuera de bloques por mucho tiempo). E-Bitcoin sabría, viendo su registro, que esas transacciones de tal bloque TUC siguen "pendientes externamente". Podría entonces tomar acción (increase fee o notificar error).
- **Integridad**: Es improbable pero si la red externa tiene un rollback (ej: un reorg mayor en Bitcoin que elimina la transacción de e-Bitcoin), la referencia en e-Bitcoin sigue ahí diciendo que se procesó. E-Bitcoin en su registro no revertiría eso (porque su chain es irreversible), pero sabría que externamente no se finalizó. Sería una situación a gestionar (tal vez un eventual remedy: or re-send or reimburse; estos son escenarios extremos).

En general, el registro primario en e-Bitcoin es la *fuente de verdad* de la intención y momento del pago, mientras que la red externa es la *fuente de verdad* de la entrega efectiva de valor. Manteniendo la referencia en e-Bitcoin, se unen ambos: se puede mapear un registro e-Bitcoin a transacción externa.

Es parecido a un sistema contable central que registra pagos bancarios: el libro central dice "pagado cheque #123 el día tal", aunque el banco es quien movió el dinero. Ambos se complementan, pero el interno es su base de control.

Esto es importante también para interoperabilidad futura y reporting: e-Bitcoin puede producir extractos para sus usuarios, enumerando todos sus pagos (internos y externos) con tiempos TUC. Los externos vendrán con, por ejemplo, un TXID externo asociado, pero la secuencia y comprobante principal es TUC.

Finalmente, desde la perspectiva conceptual: al conservar la referencia permanente en e-Bitcoin, TUC logra su promesa de eliminar doble tiempo e interpolaridad, porque todos los eventos, incluso los que concluyen en otra red, quedaron unificados en un solo timeline (TUC) con marcas de tiempo consistentes. La ejecución multi-cadena no fragmentó la cronología de la operación: siempre se puede volver al registro e-Bitcoin para ver cuándo ocurrió (y saber luego dónde se reflejó externamente).

En resumen, aunque los fondos salgan a distintos destinos, **el documento maestro de la transacción** sigue siendo la cadena e-Bitcoin. Este registro permanente por bloque TUC asegura que la historia completa de las transferencias está consolidada en un lugar, confiriendo a e-Bitcoin el rol de ledger unificador entre diversas plataformas. Es la pieza final

que cierra el círculo de interoperabilidad temporal: sin importar dónde termine la transacción, su huella temporal y su compromiso original residirán siempre en e-Bitcoin.

11. Cierre conceptual (resultado del proceso)

11.1 Por qué TUC elimina el doble tiempo

El **Tiempo Universal por Consenso (TUC)** elimina el problema del "doble tiempo" al establecer una única línea temporal acordada para todos los eventos, reemplazando la necesidad de coordinar entre múltiples referencias horarias. En sistemas tradicionales, una transacción a menudo implica dos tiempos: el tiempo en que las partes *desean* o *acuerdan* que ocurra (tiempo contractual, calendario civil) y el tiempo en que efectivamente se ejecuta en una plataforma (tiempo de procesamiento de la red, a veces impredecible). Esto obliga a llevar un doble control: por ejemplo, empresas manejan un calendario interno para pagos y a la vez lidian con los tiempos reales de transferencia bancaria o blockchain, que pueden desfasarse. Con TUC, esa duplicidad desaparece porque **el momento programado y el momento de ejecución se unifican** en el bloque TUC designado. No hay divergencia: la hora pactada *es* la hora de la transacción en el ledger, garantizada por consenso.

Al tener todos los participantes de e-Bitcoin una referencia común (los bloques TUC numerados en secuencia), se evita la asincronía entre "tiempo legal" y "tiempo de red". Como resaltaban analistas, las firmas cripto solían verse obligadas a llevar “calendarios duales” – el de las obligaciones legales y el de las liquidaciones de la blockchain. TUC resuelve esto al inscribir directamente las obligaciones en un calendario neutral aceptado por todos. Por ejemplo, si una ley exige pagar salarios el día último de mes, e-Bitcoin permite agendarlo exactamente en ese día a la hora indicada; la red asegurará que se ejecute en ese instante consensuado, eliminando desfases. Ya no existe la incertidumbre de "¿caerá la transacción hoy o mañana según la congestión?": el tiempo de la transacción es determinístico.

Además, TUC provee una **marca de tiempo de consenso** para cada evento, que sirve como punto único de verdad. Todos los nodos acuerdan cuándo ocurrió cada transacción, evitando interpretaciones diferentes. En sistemas sin TUC, podría haber disputas o confusiones: "según mi reloj te pagué a las 23:59, según el bloque salió a las 00:02 del día siguiente". Con TUC, ese dilema desaparece: se pagó en el bloque TUC X, punto final, referencia universal.

La eliminación del doble tiempo simplifica también la interoperabilidad organizacional. Empresas multinacionales, por ejemplo, que operan en distintos husos horarios pero usan TUC, ya no tienen que convertir horas ni preocuparse de días bancarios: usan directamente el tiempo TUC (neutro, alineado al consenso, no a zonas horarias). Como se señalaba en estudios recientes, Bitcoin ya ofrecía "un reloj compartido y neutral que nadie puede pausar o reiniciar"; TUC lleva ese concepto a un sistema completo de pagos programables. Todos los eventos financieros relevantes pueden ahora referirse a un solo estándar temporal inmutable, eliminando la necesidad de sincronizar manualmente agendas externas con tiempos de confirmación variables.

En suma, TUC elimina el "doble tiempo" al **colapsar el tiempo planificado y el tiempo real en una sola dimensión temporal consensuada**. Esto brinda certeza temporal: lo que antes requería conciliar dos relojes (el de la intención y el de la ejecución) ahora se resuelve en uno solo (el tiempo TUC). Como consecuencia, procesos y acuerdos se vuelven más simples, transparentes y fiables, pues "el cuándo" deja de ser una variable incierta o dual para convertirse en un dato fijo garantizado por la red.

11.2 Por qué e-Bitcoin habilita interpolaridad

La arquitectura de e-Bitcoin, cimentada en el tiempo unificado TUC, habilita la **interpolaridad** tecnológica al permitir que múltiples redes y sistemas interactúen a través de un eje temporal común. Tradicionalmente, lograr interoperabilidad entre diferentes blockchains o plataformas ha sido complejo porque cada una opera con su propia secuencia de bloques y tiempos. e-Bitcoin resuelve este escollo proporcionando un "terreno neutral" –su calendario TUC lineal– donde eventos de distintas cadenas pueden ser programados y correlacionados sin ambigüedad temporal.

Mediante TUC, e-Bitcoin actúa como un **hub temporal**: se pueden agendar transferencias hacia Bitcoin, Ethereum u otras redes todas en la misma agenda, lo que significa que, por ejemplo, un usuario podría orquestar una operación donde envía valor a una dirección de Bitcoin *y* activa un contrato en Ethereum al mismo tiempo TUC, algo prácticamente imposible de coordinar con precisión usando cada red por separado. Esta sincronización central elimina la necesidad de "adivinar" cuándo ocurrirá un bloque en otra cadena o de depender de disparadores externos, pues e-Bitcoin ya integra esas acciones en su cronograma. Dicho de otro modo, e-Bitcoin **interpela** a las demás redes en su propio idioma temporal: "haré X en Bitcoin en el bloque TUC N", y lo cumple en ese momento, luego "haré Y en Ethereum en el bloque TUC M", y así con todas, alineando esos eventos en una sola línea. Las distintas blockchain, aunque no se comuniquen directamente entre sí, quedan **interpoladas temporalmente** a través de e-Bitcoin.

Esto habilita casos de uso de interoperabilidad avanzada. Por ejemplo:

- **Pagos atómicos inter-cadena**: usando TUC, se puede programar que un pago en cadena A ocurra a la par que otro en cadena B, logrando efecto similar a un atomic swap pero coordinado por tiempo en lugar de hash locks. Ambos pagos se ejecutan en el mismo segundo en sus respectivas cadenas, minimizando riesgo de desfase.
- **Cadena de auditoría unificada**: empresas que manejan activos en varias blockchains pueden usar e-Bitcoin para consolidar su actividad: cada movimiento externo queda registrado en TUC, facilitando una auditoría integral. La interpolaridad se ve en que eventos en distintas cadenas se ordenan cronológicamente en e-Bitcoin, permitiendo responder preguntas como "¿qué pasó primero, la operación en Ethereum o en Fabric?" sin dudas, pues TUC lo establece.
- **Orquestación multi-plataforma**: Por ejemplo, disparar un pago en Bitcoin y una notificación en un sistema legacy al mismo tiempo. E-Bitcoin maneja el pago y puede notificar mediante algún riel integrado (o al menos queda el timestamp para correlacionar).

En definitiva, e-Bitcoin logra la interpolaridad al servir de **capa de coordinación** entre sistemas heterogéneos. Su tiempo unificado es la clave: sin un común denominador temporal, cada red va a destiempo. Con él, se puede entrelazar eventos de diferentes orígenes en secuencia. Esto era un obstáculo señalado en la industria: la dificultad de alinear operaciones cross-chain precisamente por la falta de un tiempo común. TUC viene a ser ese estándar neutral que, como se dijo de Bitcoin, es un "reloj neutral que no depende de terceros", ahora extendido a un servicio completo.

Por tanto, e-Bitcoin habilita la interpolaridad porque **provee el pegamento temporal que une sistemas dispares**. Cada red mantiene sus reglas, pero todas se integran vía e-Bitcoin que las coordina en el tiempo. Esto reduce fricción: en lugar de complejos protocolos de sincronización, basta confiar en TUC como agenda global. Tecnológicamente, e-Bitcoin se coloca como una *meta-capa* sobre las blockchains existentes, no reemplazándolas, sino sincronizándolas – un logro antes no alcanzado debido a la falta de una referencia temporal compartida y confiable, que TUC ahora aporta.

11.3 Resumen operativo: agenda → entrada → umbral → merkle → distribución → registro

Recapitulando el funcionamiento integral del sistema TUC en e-Bitcoin, podemos delinear el flujo operativo en seis etapas secuenciales:

1. **Agenda (Planificación)** – Todo inicia con la planificación en la agenda unificada TUC. Los usuarios seleccionan en el calendario consensuado la fecha y bloque deseados para sus transferencias, definiendo también el riel (destino de red) y pudiendo añadir detalles como HashPaper. Queda así **agendado** el evento en el tiempo consensuado, con posibilidad de confirmación mutua si es requerido. *(Ejemplo: se agenda un pago para el bloque del 30 de junio 2026 a las 15:00 TUC, riel PAY-BTC, monto X BTC.)*

2. **Entrada (Ingreso de solicitud)** – La orden agendada o en tiempo real **entra** al sistema e-Bitcoin. Si es agendada, se almacena para el bloque futuro correspondiente; si es inmediata, se asigna al bloque en curso según capacidad. Se sella con el timestamp de entrada y queda pendiente de procesamiento. *(Ej.: la orden entra el 15 de junio a la 10:00 TUC como "pago X BTC a Y el 30/06 15:00", confirmada por receptor el 20 de junio.)*

3. **Umbral (Procesamiento)** – Al llegar el **umbral del presente** del bloque agendado, la transferencia cruza a estado **procesado**. En ese momento, e-Bitcoin ejecuta la transacción: debita y acredita saldos internos, sella la operación en el bloque TUC correspondiente y la marca como irreversible. Si era cancelable y no se canceló hasta antes de este punto, ahora ya no hay marcha atrás. *(Ej.: al marcar las 15:00:00 TUC del 30 de junio 2026, el pago se procesa en e-Bitcoin: se descuenta de la cuenta del pagador y queda registrado para enviarse a la dirección Bitcoin destino.)*

4. **Merkle (Consolidación)** – Inmediatamente después del procesamiento, e-Bitcoin agrupa las transacciones del bloque para integridad y entrega. Construye el **Merkle**

interno con todas las transferencias internas del bloque, obteniendo una raíz que asegura la integridad del bloque. Paralelamente, para cada riel externo, recopila las transferencias externas de varios bloques TUC si es necesario (multi-bloque) y forma un **Merkle por red** que resume esas operaciones. Esto genera paquetes listos para la salida. *(Ej.: el pago X BTC forma parte del Merkle de riel BTC junto con otras transferencias a Bitcoin ocurridas en esos 10 min. Se obtiene una raíz Merkle representando, digamos, 50 pagos a Bitcoin.)*

5. **Distribución (Hacia redes externas)** – e-Bitcoin actúa como **distribuidor**. Según cada riel, **enruta** los paquetes de transacciones externas al lobby correspondiente. Cada lobby (Bitcoin, Ethereum, etc.) recibe sus transacciones y las inserta en la red externa según las reglas de ésta. La red externa las procesa en sus bloques (minado, consenso propio) y las confirma. Si la red externa no admite paquetes Merkle directos, el lobby envía transacciones individuales; si admite compromisos, se aprovecha la preconstrucción. *(Ej.: e-Bitcoin envía una transacción Bitcoin agregada con X BTC a Y (y otras salidas) al mempool de Bitcoin. Los mineros de Bitcoin la incluyen; ~10 minutos después queda confirmada en un bloque de Bitcoin.)*

6. **Registro (Referencia permanente)** – Finalmente, e-Bitcoin conserva en su **registro contable interno** el rastro completo del evento. El bloque TUC donde se procesó la transacción permanece como referencia permanente, vinculando el evento a cualquier identificación externa (p.ej., TXID de Bitcoin). Este registro unificado permite auditoría y seguimiento global. *(Ej.: en e-Bitcoin queda para siempre la anotación: "Bloque TUC 2026-06-30 15:00 – Pago de X BTC a addr Y – Procesado, TxID externo abc123... en Bitcoin".)*

Este flujo – **agenda** → **entrada** → **umbral** → **merkle** → **distribución** → **registro** – demuestra la sólida orquestación de e-Bitcoin. La **agenda** garantiza planificación precisa; la **entrada** asegura admisión ordenada; el **umbral** marca la ejecución fiel en tiempo; el **merkle** brinda integridad y empaquetamiento; la **distribución** logra la interoperabilidad entregando los valores a cada red; y el **registro** cierra el ciclo dejando un historial unificado. Cada etapa es crítica y se encadena con la siguiente para proporcionar un proceso completo, confiable y transparente.

En conclusión, e-Bitcoin con TUC ofrece una solución integral a problemas históricos en sistemas transaccionales: sincroniza tiempos, integra cadenas diversas y documenta todo en un solo libro mayor temporal. El resultado es un funcionamiento *lineal, interoperable y certificado* del movimiento de valor y la ejecución de compromisos a través del tiempo unificado, cumpliendo así la promesa planteada en la definición y alcances iniciales del sistema.