Taller Blockchain

Diseño, desarrollo e instalación de contratos inteligentes

Dr. Iván S. Razo-Zapata
Dr. Alberto F. Martínez

Día 5, Miércoles 10 de Marzo, 2021

Agenda

- Contratos inteligentes
- Instalación y pruebas

Diseño, desarrollo e instalación de contratos inteligentes

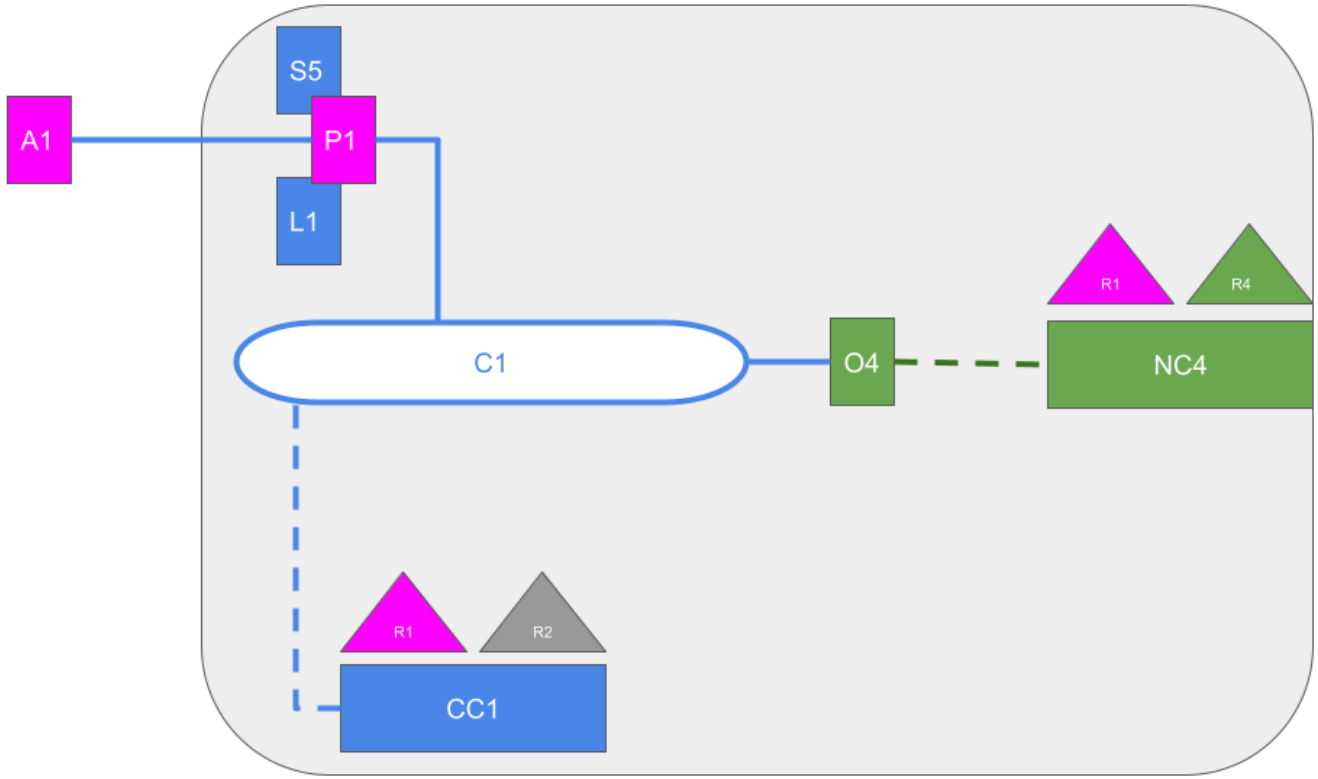

Figure: Acceso de una aplicación a un peer, Parte 1. Figura basada en hyperledger-fabricdocs Documentation.

Diseño, desarrollo e instalación de contratos inteligentes

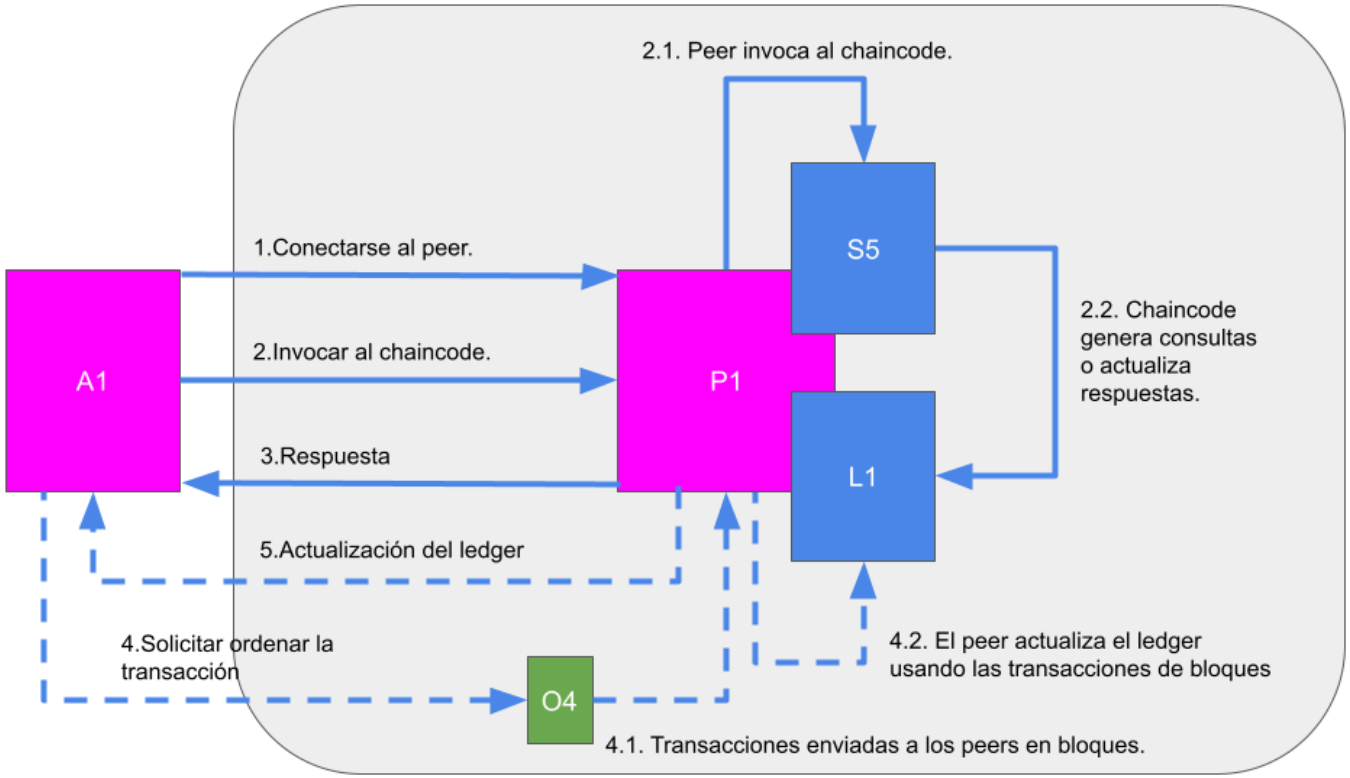

Figure: Acceso de una aplicación a un peer, Parte 2. Figura basada en hyperledger-fabricdocs Documentation.

Diseño, desarrollo e instalación de contratos inteligentes

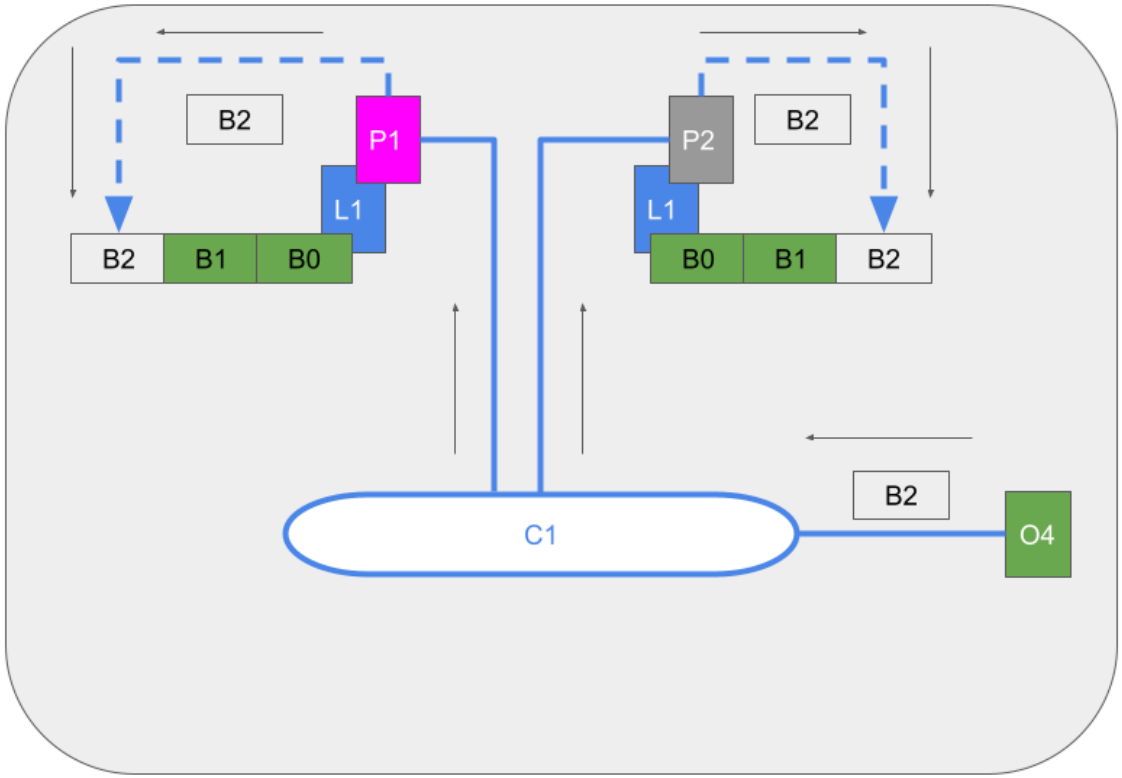

Figure: Como se actualiza el ledger. Figura basada en hyperledger-fabricdocs Documentation.

Diseño, desarrollo e instalación de contratos inteligentes

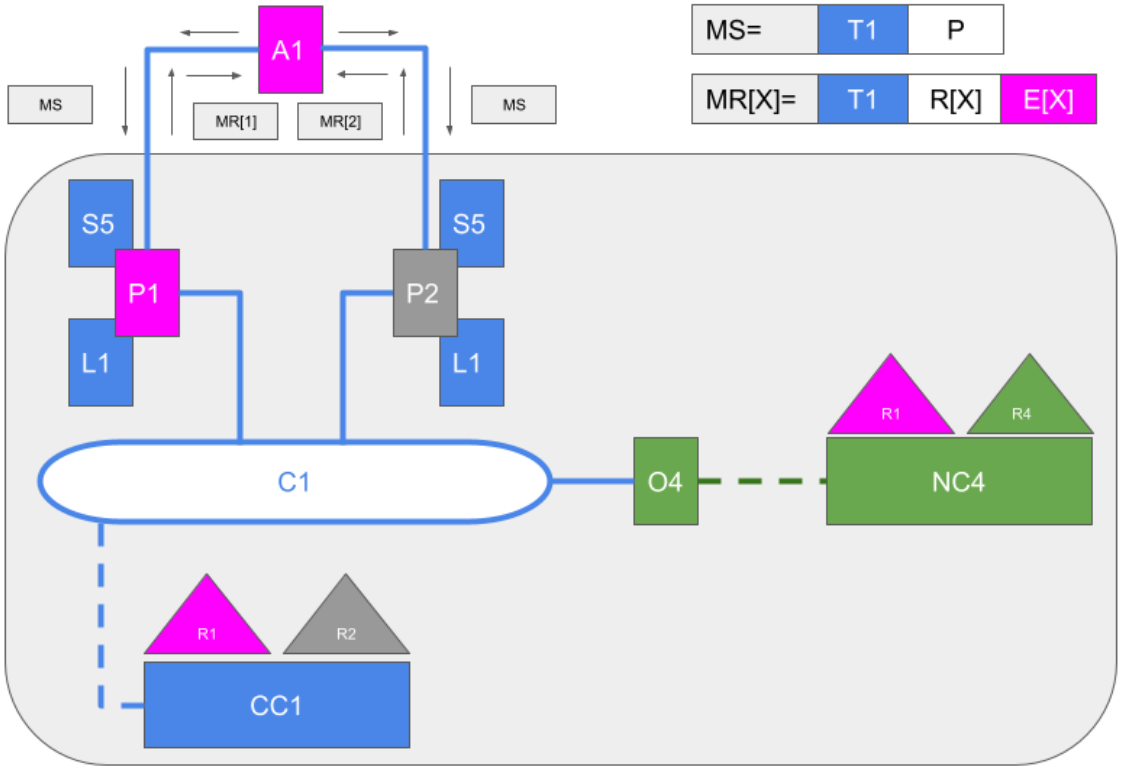

Figure: Como entra una propuesta a cada peer. Figura basada en hyperledger-fabricdocs Documentation.

Diseño, desarrollo e instalación de contratos inteligentes

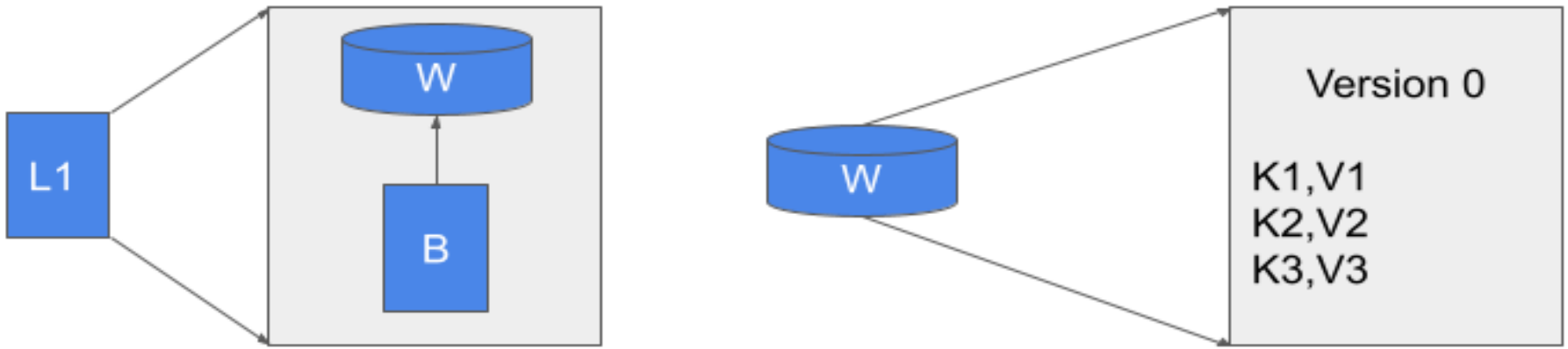

Figure: Componentes de un ledger. Figura basada en hyperledger-fabricdocs Documentation.

Diseño, desarrollo e instalación de contratos inteligentes

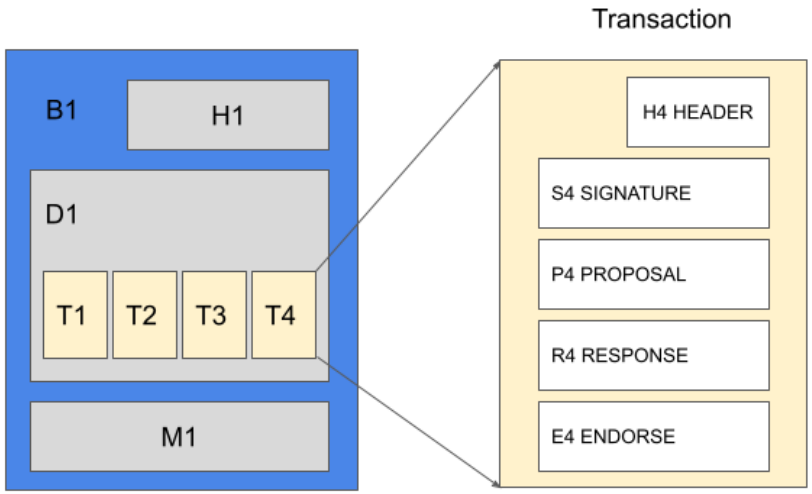

Figure: Como se almacenan las transacciones. Figura basada en hyperledger-fabricdocs Documentation.

Diseño, desarrollo e instalación de contratos inteligentes

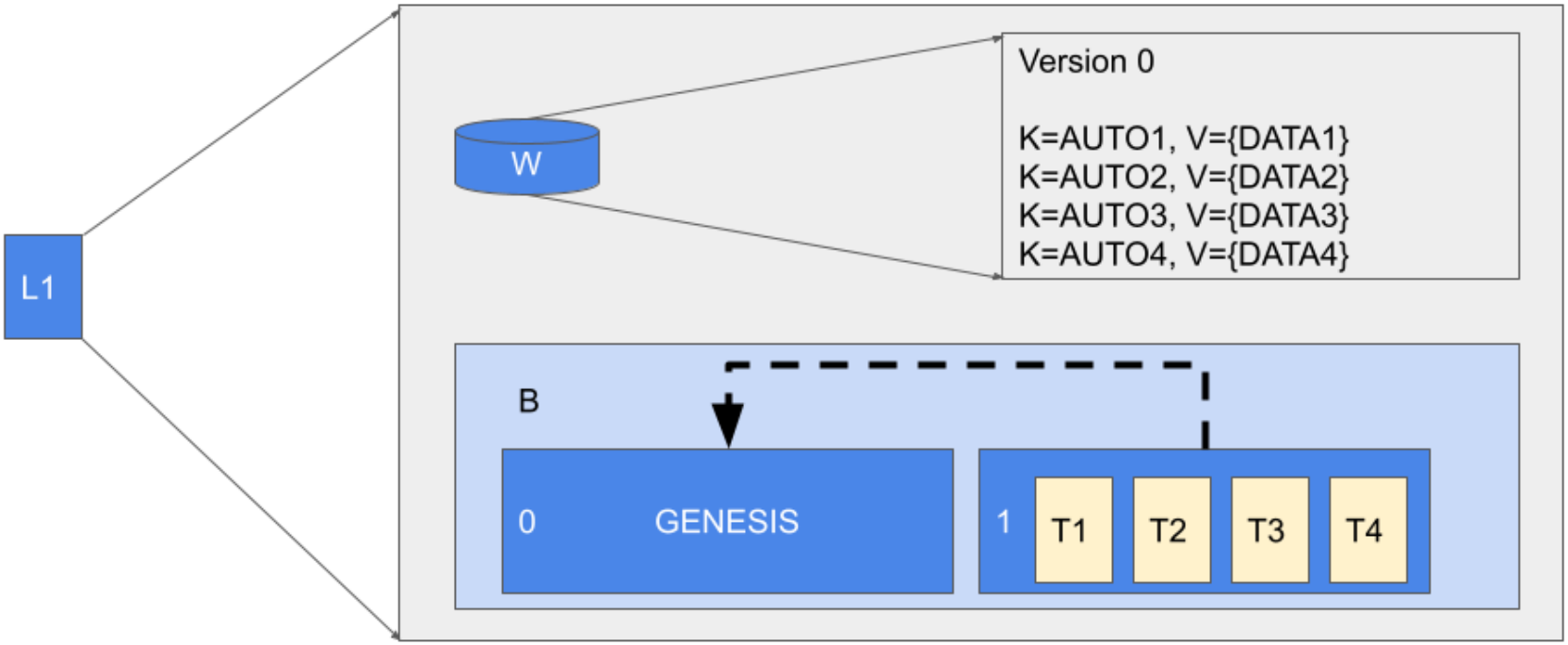

Figure: Ledger y cadena de bloques. Figura basada en hyperledger-fabricdocs Documentation.

Diseño, desarrollo e instalación de contratos inteligentes

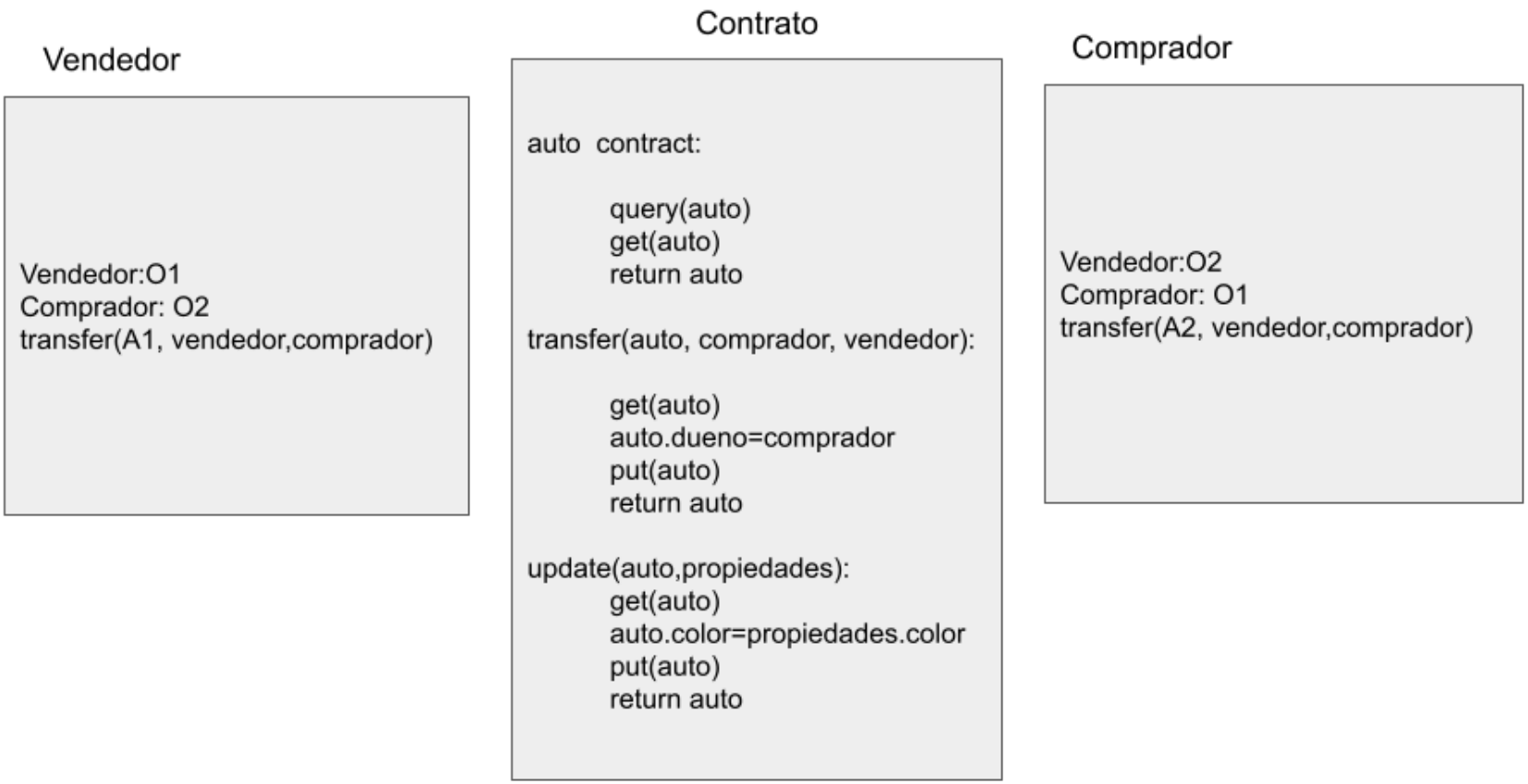

Figure: Ejemplo de contrato. Figura basada en hyperledger-fabricdocs Documentation.

Diseño, desarrollo e instalación de contratos inteligentes

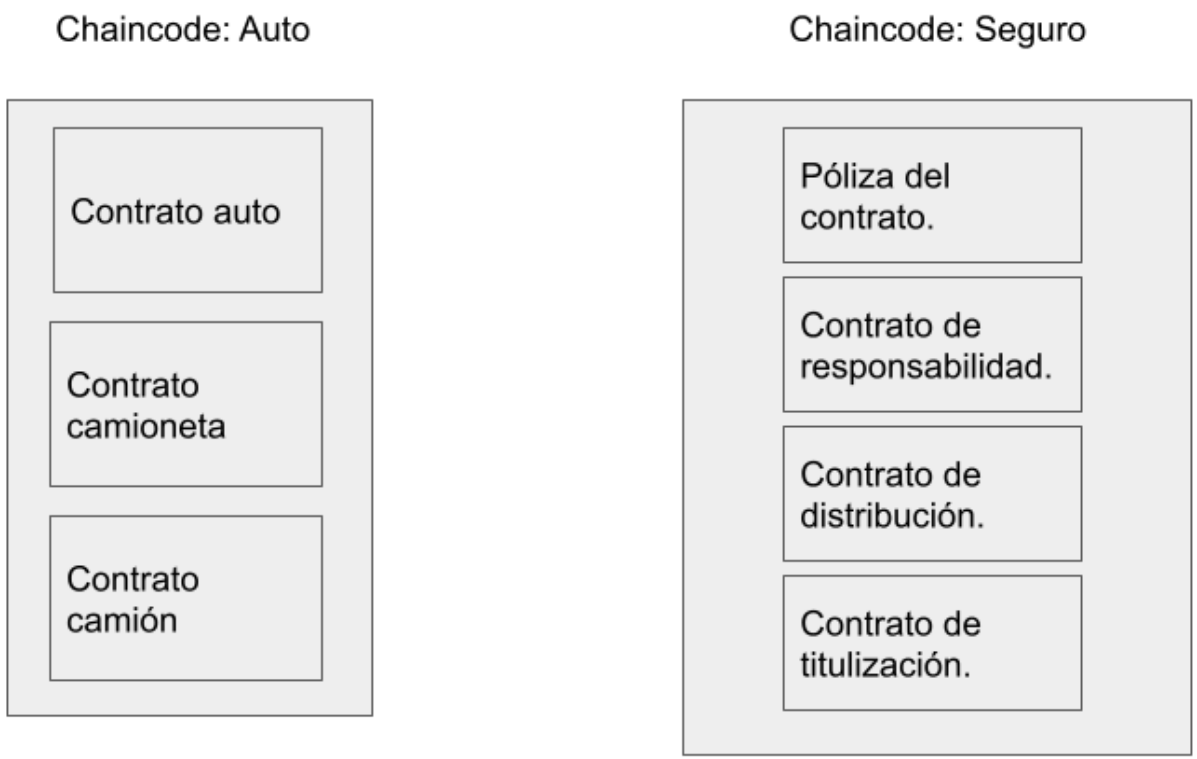

Figure: Tipos de contrato. Figura basada en hyperledger-fabricdocs Documentation.

Diseño, desarrollo e instalación de contratos inteligentes

Formato: El smart contract se define en algún lenguaje de programación, como Golang, JavaScript o Java. Por ejemplo, siguiendo con el ejemplo de auto, tenemos:

```
async crearAuto(ctx, NoAuto, make, modelo, color, propietario) {
        const car = {
                color,
                docType: 'auto',
                make,
                modelo,
                propietario,
        };
        await ctx.stub.putState(NoAuto, Buffer.from(JSON.stringify(auto)));
}
```

Algorithm 1: Ejemplo de chaincode, basado en hyperledger-fabricdocs Documentation.

Diseño, desarrollo e instalación de contratos inteligentes

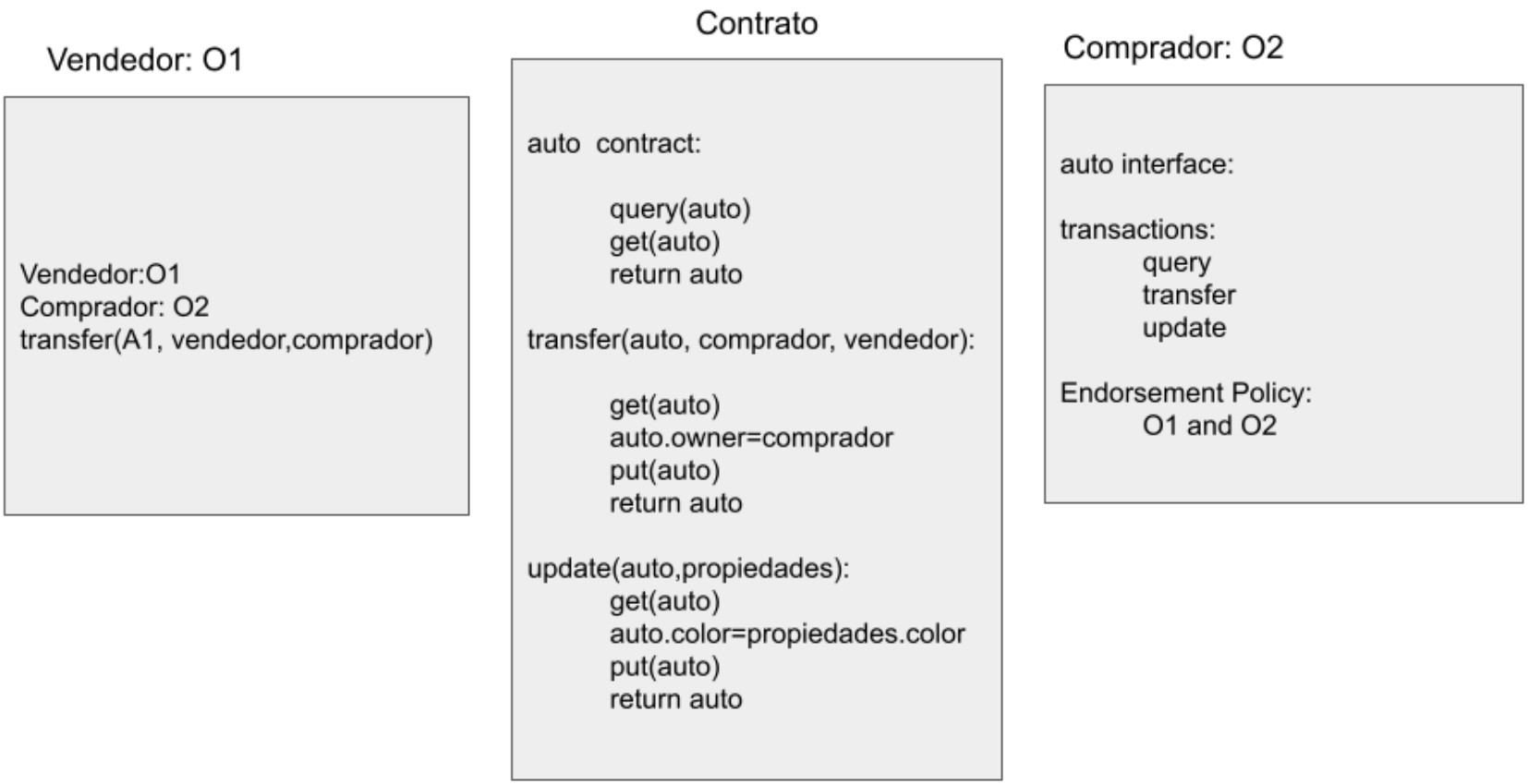

Figure: Endorsement. Figura basada en hyperledger-fabricdocs Documentation.

Diseño, desarrollo e instalación de contratos inteligentes

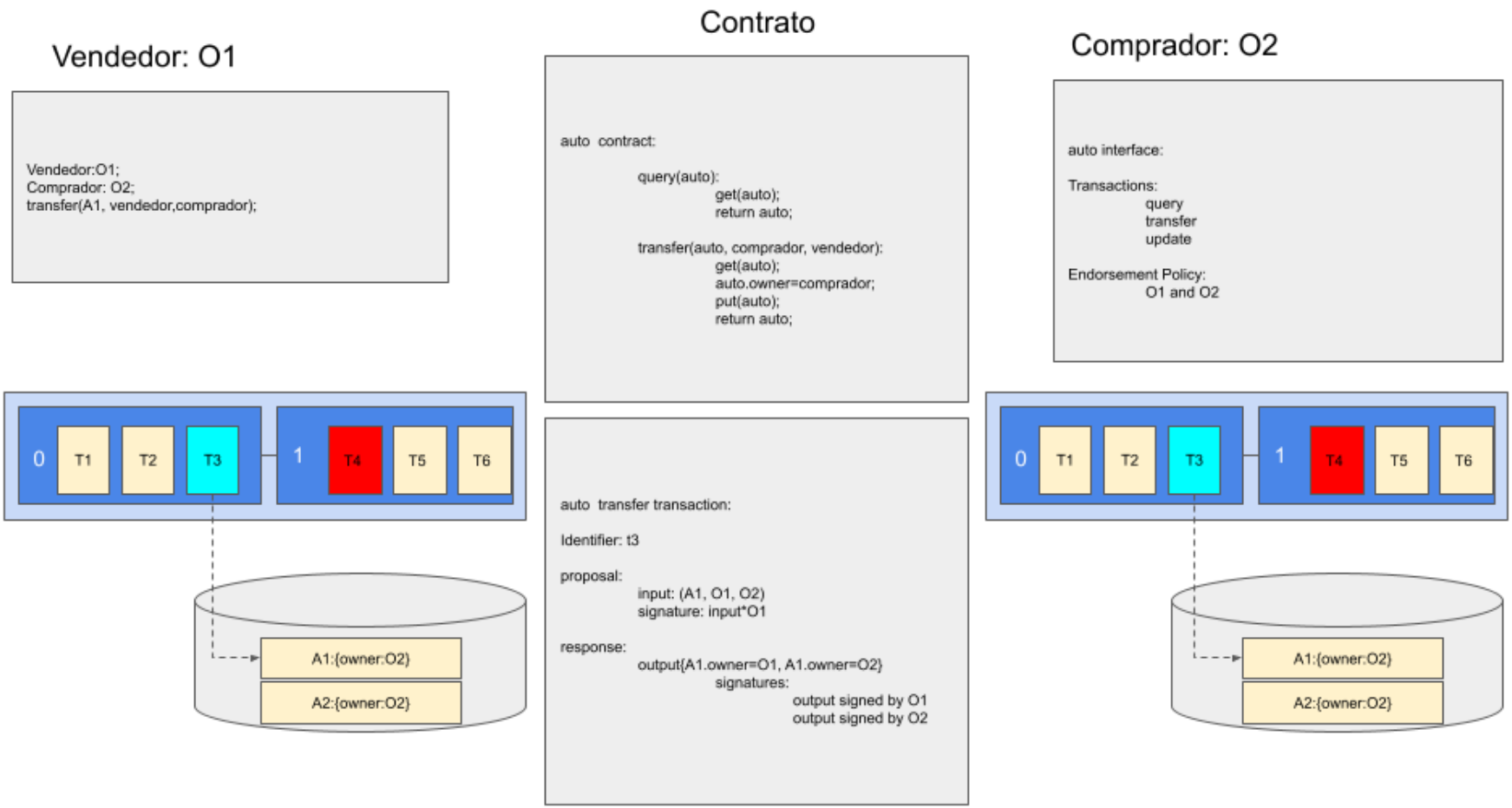

Figure: Validación. Figura basada en hyperledger-fabricdocs Documentation.

Diseño, desarrollo e instalación de contratos inteligentes

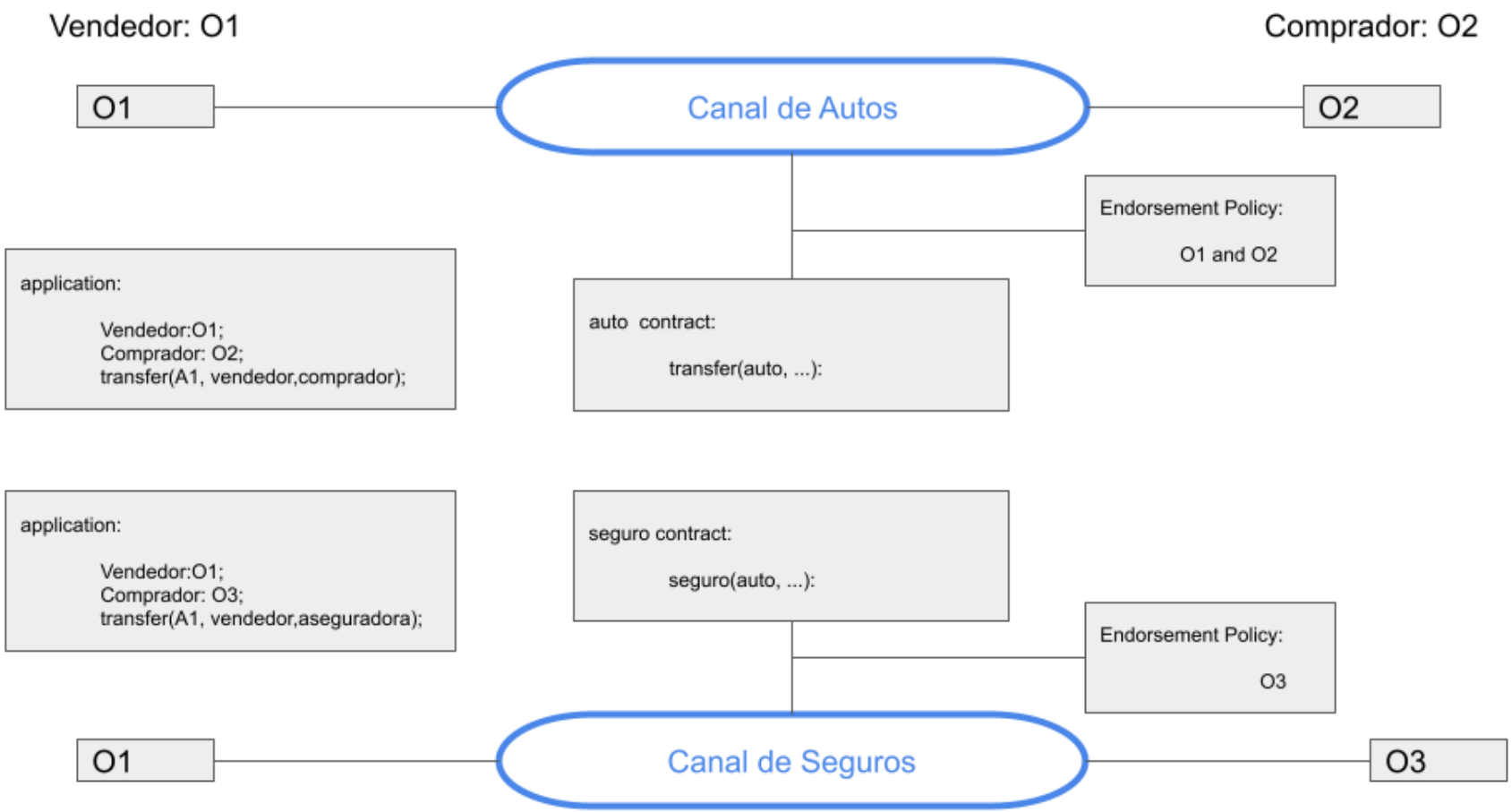

Figure: Contratos en 2 canales. Figura basada en hyperledger-fabricdocs Documentation.

Diseño, desarrollo e instalación de contratos inteligentes

Ciclo de vida del chaincode. Está compuesto de los siguientes 4 pasos:

- Empaquetar el chaincode (sinónimo de Smart Contract).
- Instalar el chaincode en cada nodo (peer) de la organización que lo requiera.
- Proceso de aprobación del chaincode por cada organización. Se requiere la mayoría de la aprobación de las organizaciones participantes.
- Hacer commit de la definición del chaincode sobre el canal. Una organización recopila los respaldos de acuerdo a la política del canal usado, posteriormente envía la transacción y aplica el proceso de commit para definir el chaincode.

Diseño, desarrollo e instalación de contratos inteligentes

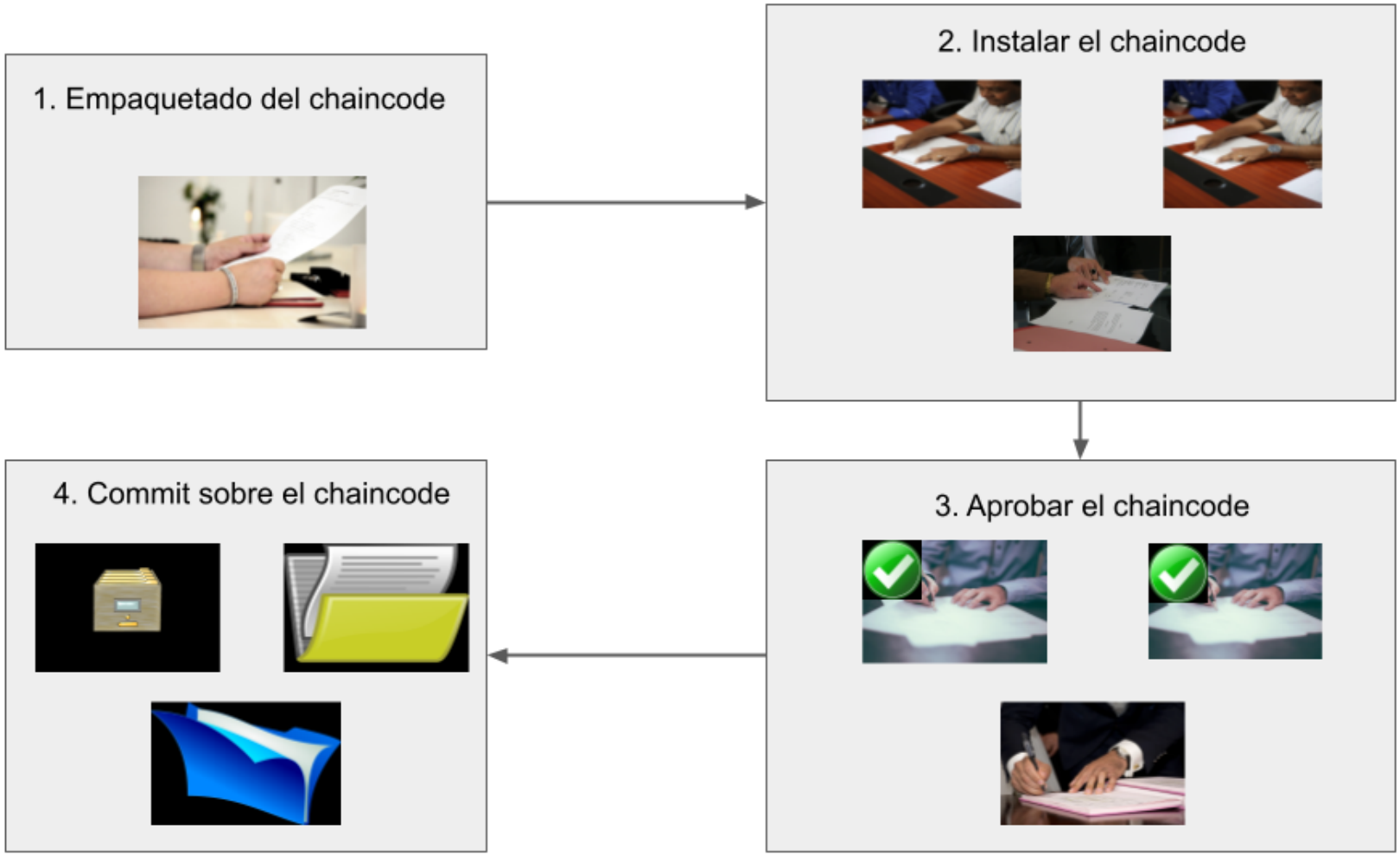

Figure: Ciclo de vida de chaincode, ejemplo análogo en el mundo real. Imágenes tomadas de `https://pixabay.com/`.

Diseño, desarrollo e instalación de contratos inteligentes

Empaquetar el chaincode. Al empaquetar el chaincode, se tiene lo siguiente:

- Ser empaquetado en un archivo tar.gz.
- El paquete contiene 2 archivos, el chaincode y un archivo de referencia en formato json.
- El archivo de metadatos se identifica como 'Chaincode-Package-Metadata.json'. Allí se especifica la ruta, el lenguaje en el que se hizo y la etiqueta del paquete.

Para nuestro caso de ejemplo, lo llamaremos fabcar.tar.gz.

Diseño, desarrollo e instalación de contratos inteligentes

Figure: Primer paso del ciclo de chaincode, empaquetado. Figura basada en hyperledger-fabricdocs Documentation.

Diseño, desarrollo e instalación de contratos inteligentes

Instalar el chaincode. Después de ser recibido por cada uno de los peers, el chaincode se desempaqueta, se instala y se construye. Se recomienda empaquetar una sola vez, para evitar algún conflicto. Si la (re)construcción ha sido exitosa, se obtendrá un hash como ID para ese chaincode.

Diseño, desarrollo e instalación de contratos inteligentes

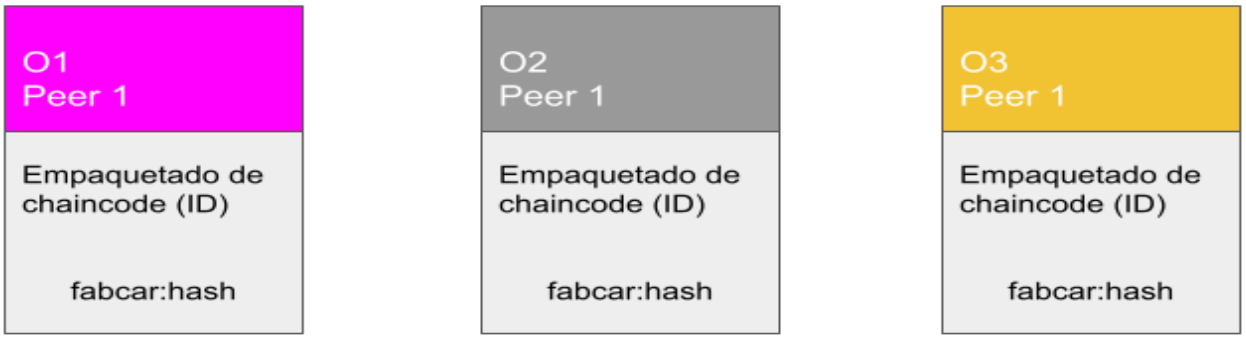

Figure: Segundo paso del ciclo de chaincode, instalación. Figura basada en hyperledger-fabricdocs Documentation.

Diseño, desarrollo e instalación de contratos inteligentes

Aprobar el chaincode. La definición del chaincode deberá contener (entre otros campos):

- Nombre. Como las aplicaciones lo identificarán para utilizarlo.
- Versión. Llevar el control de la versión del chaincode (actualizaciones)
- Secuencia. Para tener el control de las actualizaciones (instalación=1, actualización=2).
- Política de aprobación. Por default, la mayoría debe aprobar.
- ID del empaquetado.

Diseño, desarrollo e instalación de contratos inteligentes

Aprobar el chaincode. Después de ser aprobado por la organización, dicha aprobación debe ser enviada al Ordering Service a través del Administrador designado por la organización en la red. Posteriormente el Ordering Service es quien distribuirá el chaincode, el cual estará a disposición de los peers.

Diseño, desarrollo e instalación de contratos inteligentes

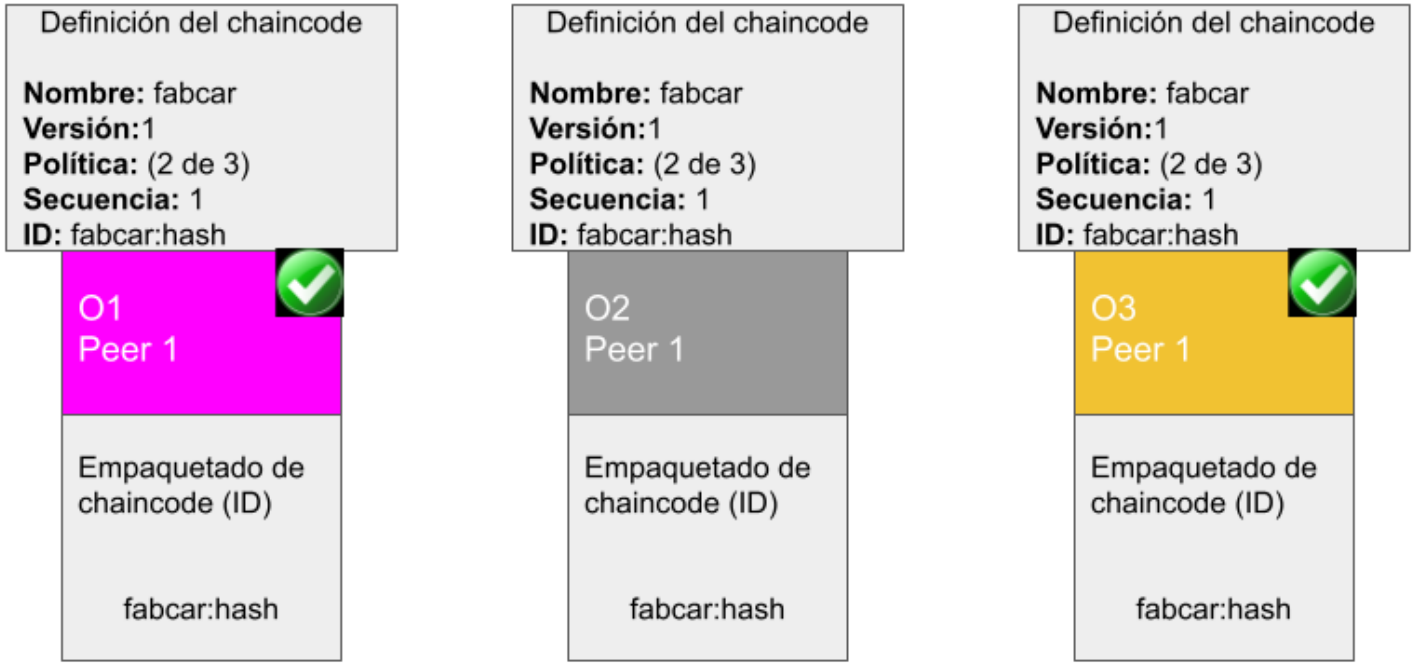

Figure: Tercer paso del ciclo de chaincode, aprobación. Figura basada en hyperledger-fabricdocs Documentation.

Diseño, desarrollo e instalación de contratos inteligentes

Proceso de commit. Una organización (el que está definido como administrador) es la que realiza el proceso de commit en el canal y la correspondienten definición del chaincode. Dicha organización consulta a los peers con el chaincode aprobado y cada uno verifica si su organización lo aprobo. Si así fue, la transacción es enviada al Ordering Service para posteriormente ser hecho el proceso de commit en el canal utilizado.

Diseño, desarrollo e instalación de contratos inteligentes

Proceso de commit. Se requiere satisfacer la política de LifecycleEndorsement para que proceso de commit se pueda llevar a cabo. Por lo general, nuevamente se requiere que la mayoría apruebe. Pero dicha política puede ser modificable.

Diseño, desarrollo e instalación de contratos inteligentes

Proceso de commit. Una marcada diferencia es que aun si una organizacion no usa el chaincode, puede que se necesite su aprobación para que el LifecycleEndorsement se cumpla y se haga el proceso de commit. Ya que el proceso de commit se haya realizado, los peers ya pueden utilizarlo.

Diseño, desarrollo e instalación de contratos inteligentes

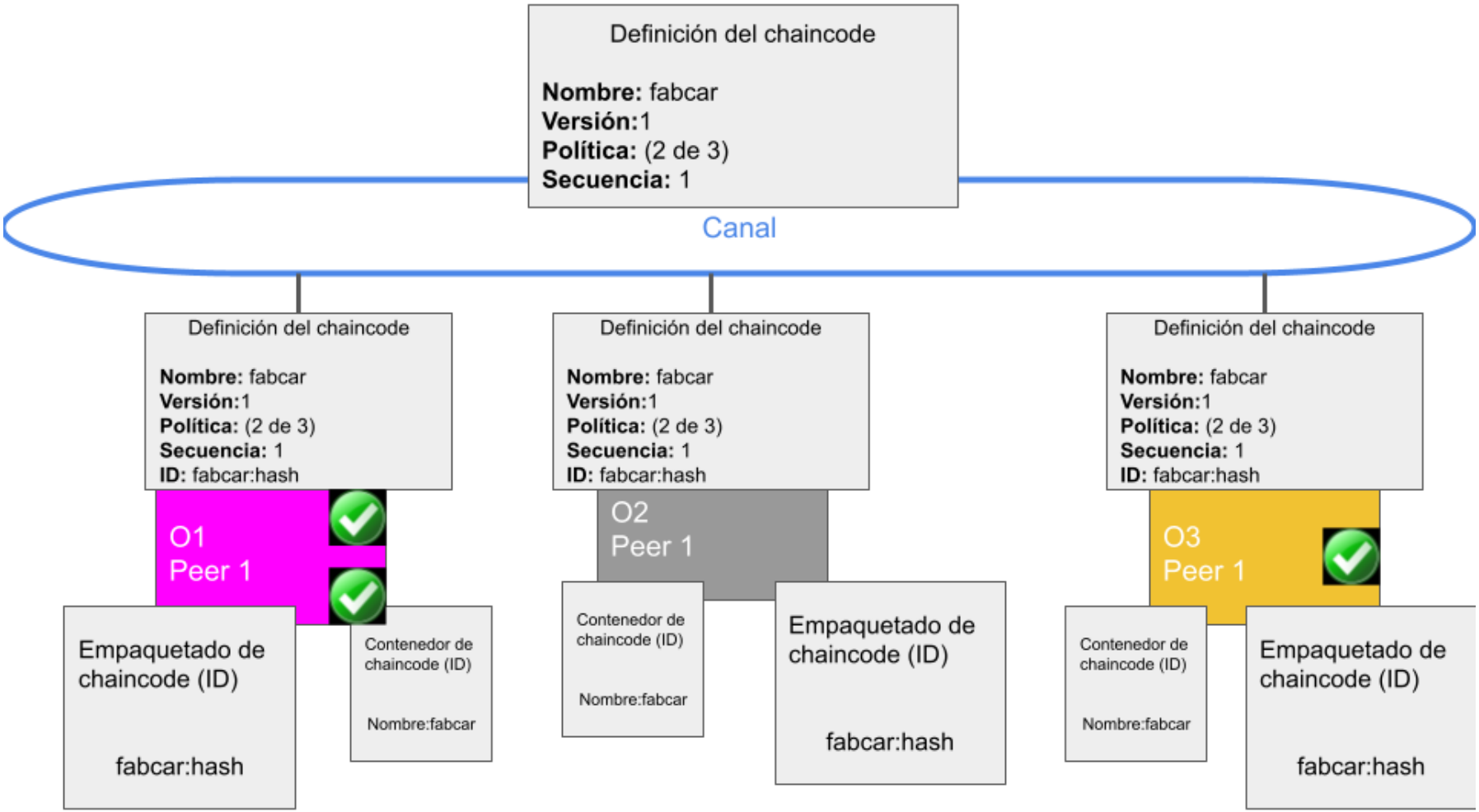

Figure: Cuarto paso del ciclo de chaincode, commit. Figura basada en hyperledger-fabricdocs Documentation.