

Aşılamayan Bütçe: Yönetişim Noterlerinden Yönetişim Mekanizmalarına

Şen Ferhat
Bağımsız araştırmacı
sen@kycel.com

Temmuz 2026, çalışma taslağı v0.1
(İngilizce aslından çeviri; asıl metin: `main.tex`)

Özet

Zincir üstü yönetim sistemlerinin büyük çoğunluğu birer *noter*dir: kararları kaydeder, ancak icra etmezler. Onaylanan bir teklif bir olay (event) yayar; yürütme, çevredeki altyapıyı her kim işletiyorsa ona bırakılmıştır. Bu çalışmada, kurumları, devletleri, şirketleri, kooperative’leri, federasyonları, 25 alan-nötr akıllı sözleşme primitifinin (siciller, meclisler, hazineler, bütçeler, zaman kilitleri, tesciller, iş emirleri, tahkim) kompozisyonları olarak modelleyen STOA çerçevesini sunuyoruz: yaklaşık 30 Solidity sözleşmesi, toplam 2.808 satır kod; görsel süreç haritası, dağıtım manifestosu ve çalışan sistemin üçü de tek bir graftan türer. Katkılarımız: (1) *noter/mekanizma* ayrımı, her bileşeni ya *öz-icralı* (self-enforcing; kuralı zincir üstünde fiziksel olarak ihlal edilemez) ya da *beyana dayalı* (attested: zincir dışı gerçekliğe ilişkin noterlenmiş bir iddia) olarak etiketleyen bir dürüstlük taksonomisiyle işlevselleştirilir; 25 primitifin 16’sı öz-icralıdır ve etiket, kullanıcıya dönük tasarım yüzeyine ve dağıtım manifestosuna taşınır; (2) bir meclis oylamasının, oylanan zincir-üstü eylemi itiraz-pencereli bir zaman kilidi üzerinden *zorunlu kıldığı* bir *aktüasyon döngüsü* ve dağıtım anında gerçekleşen, sonrasında dağıtıcının (deployer) kendi yarattığı sistem üzerindeki denetimini kanıtlanabilir biçimde yitirdiği bir *yönetişim devri*; ve (3) *kod-olarak-bütçe* (budget-as-code): oyla onaylanan ödenekler, hazinenin kendisi tarafından her çıkışta danışılan bir harcama politikası olarak icra edilir; bütçeyi aşmak bir kural ihlali değil, bir işlem geri dönüşü (revert) hâline gelir, hazinenin sahibi için bile. Üç katkıyı, dağıtıcının her bypass girişiminin geri döndüğü, aynı eylemin ise yalnızca ve yalnızca anayasal yolu, teklif, oy, nisap, zaman kilidi, yürütme, izlediğinde başarıya ulaştığı, yeniden üretilebilir bir vaka çalışmasıyla gösteriyoruz. Kodlanamayanların, kimlik, oracle’lar, gizlilik ve hukuki tanınma, dürüst bir dökümüyle bitiriyoruz.

1 Giriş

Akıllı sözleşmelerle yönetim, programlanabilir blokzincirlerin ortaya çıkışından beri ilkesel olarak mümkündür [1, 2]; “kod kanundur” ise daha da uzun süredir hem bir vaat hem bir uyarı olarak dolaşımdadır [3, 4, 5]. Ne var ki pratikte, dağıtılmış hâldeki “yönetişim”in ezici çoğunluğu oy *kaydından* ibarettir: teklif bir metindir, onay bir olay kayıdır ve sonrasında olan her şey, bir ödeme, bir rol değişikliği, bir parametre güncellemesi, pekâlâ oysuz da hareket edebilecek bir yönetici, bir bot ya da bir multisig tarafından gerçekleştirilir. Bu tür sistemlere *yönetişim noteri* diyoruz. Noter faydalıdır: neyin kararlaştırıldığının kurcalanmaya karşı korunaklı bir tarihçesini üretir. Ama kararları sonuçlara bağlamaz. Karar ile icrası, güvenilmesi gereken bir insan ya

da operasyon takdiri katmanıyla ayrı kalır, ki kurumların ezelden beri dizginlemekte zorlandığı katman tam da budur.

Bu makale, genel bir kurum sınıfı için bu açığın kapatılmasına, ve kapatılamadığı yerde bunun dürüstçe söylenmesine, dairdir. STOA'yı tanıtıyoruz: bir kurumun *sözleşmeler ve kablo-lar* grafi olarak belirtildiği bir çerçeve. Tasarımın her düğümü, küçük ve alan-nötr bir primitif kütüphanesinden alınan, dağıtılabilir bir Solidity sözleşmesidir; her kablo ise iki sözleşme arasında bir olaydan-metoda bağlantıdır. Aynı graf eşzamanlı olarak (a) tasarımcının düzenlediği görsel süreç haritası, (b) dağıtım manifestosu ve (c) dağıtımdan sonra canlı sistemin kendisidir, gerçekte çalışandan sapabilecek ayrı bir “model” yoktur.

Bu çerçeve içinde üç katkı geliştiriyoruz:

K1, Noter/mekanizma ayrımının işlevselleştirilmesi. Kuralları *öz-icralı* (♦) olan bileşenleri, kural, sözleşmenin kendi denetimindeki durum üzerinde sözleşme kodunda sınanır; dolayısıyla ihlal yasak değil *imkânsızdır*: ihlal eden işlem geri döner, zincir dışı gerçekliğe dair, atanmış bir tarafın öne sürdüğü bir iddiayı (“fatura banka havalesiyle ödendi”, “teslim edilen iş kabul edilebilir”) noterleyen *beyana dayalı* (◇) bileşenlerden ayırıyoruz. Bu ayrım bir dipnot değildir; her primitif üzerinde birinci sınıf bir alan olarak taşınır, tasarım aracında ve dağıtım manifestosunda gösterilir, böylece kullanıcı, kurumunun hangi kurallarının çiğnenebilir, hangilerinin çiğnenemez olduğunu her an bilir. Mevcut kütüphanedeki 25 primitifin 16’sı öz-icralı, 9’u beyana dayalıdır. Birkaç sözleşme her iki yarıyı bilinçli olarak tek gövdede içerir; bu da sınırı somutlaştırır (§3.3).

K2, Yönetişim devri ile tamamlanan bir aktüasyon döngüsü. Meclis primitifi, zincir-üstü *eylemler* (hedef adresler ile çağrı verileri) taşıyan teklifler kabul eder. Bir teklif, bir nisap kuralı, baz puan cinsinden bir onay eşiği ve varoluşsal kararların nitelikli çoğunluk gerektirmesini sağlayan isteğe bağlı kategori-bazlı *kademeli* eşikler altında, geçtiği anda, onaylanan eylemler, gecikmesi bir *itiraz penceresi* işlevi gören bir zaman kilidine kuyruklanır; pencere dolduktan sonra bunları herhangi biri yürütebilir. Bağlanmış bir hakem (arbiter), kuyruktaki bir işlemi dava görülürken askıya alabilir, sonra hükümle sürdürebilir ya da iptal edebilir: itiraz penceresinin dışları vardır. En önemlisi, dağıtım hattı bir *yönetişim devri* ile biter: her çipin yönetim hakları dağıtıcıdan yöneten bir çipe (tipik olarak meclisin beslediği zaman kilidine) devredilir ve ortaya çıkan kontrol grafi manifestoya işlenir. Devirden sonra sistem kendi üzerinde egemendir: her doğrudan müdahalesi geri dönen bir dağıtıcı gösteriyoruz (§6).

K3, Kod-olarak-bütçe. Yönetişim tarafında değil, *hazine tarafında* çalışan bir bütçe primitifi tanıtıyoruz: hazine, *her* çıkış yolunda, kendi sahibinin başlattıkları dahil, bütçeye danışır. Meclisçe onaylanan bir ödenek bu yüzden bir maliye birimine verilen talimat değildir; kategori ve mali dönem başına, hazinenin çıkarabileceğinin fiziksel dış sınırıdır. Yaşayan bir ödenek kalemi olmadan yapılan ya da kalemin bakiyesini aşan bir harcama geri döner. Bildiğimiz kadarıyla bu, alışılmış düzeni tersine çevirir: standart governor deseninde yönetim sözleşmesi *kendi* eylemlerini kapılar, hazine ise operatörlerince harcanabilir kalır; burada reddeden hazinenin kendisidir. Kodla kurulmuş bir kurumun, kâğıt üzerindeki bir kurumun yapamayacağı şeye dair en berrak gösterimi budur: *bütçesini aşamayan bir hükûmet*.

Bu mekanizmaları yeniden üretilebilir bir vaka çalışmasında değerlendiriyoruz (§6): asgari bir devlet, sicil, senato, zaman kilidi, hazine, bütçe, dağıtılıp devredilir; dağıtıcının ve bütçesiz harcamacıların altı bypass girişimi beklenen gerekçelerle geri döner; aynı ödenek daha sonra

anayasal yoldan başarıyla geçer. Çerçeve, sözleşmeler, testler ve vaka çalışması sistemleri çalışan bir artefakt olarak mevcuttur.¹

Sınırlar konusunda da aynı ölçüde açığız (§7). Yukarıdakilerin tümü *alt katmana göreli* geçerlidir: operatörün denetlediği bir zincirde “sözleşme X’i garanti eder”, “operatör X’e söz verir”e çöker; bu yüzden tek-operatörlü dağıtımları dürüstçe *simülasyon modu* olarak konumlandırıyoruz. Mevcut mecliste oylar açıktır; bir-adres-bir-oy, henüz var olmayan bir kimlik katmanını varsayar (bir anahtar çifti bir kişi değildir) ve buradaki hiçbir şey hukuki tanınma sağlamaz. K1’in taksonomisi, kısmen, tam da bunları kesin biçimde söyleyebilme disiplini.

2 İlgili Çalışmalar

DAO çerçeveleri ve governor deseni. Yürütülebilir teklifler token yönetiminde yerleşik pratiktir: Compound tarzı governor’lar ve bunların OpenZeppelin genellemesi, bir teklifin hedefler ile çağrı verileri taşımasına, bir zaman kilidi üzerinden kuyruklanmasına ve zincir üstünde yürütülmesine izin verir [6]. Aragon ve ardılları DAO dağıtımını ve yetki yönetimini paketler [7]; DAO tasarım uzayı da iyi haritalanmıştır [8, 9, 10, 11]. STOA üç noktada ayrışır. Birincisi kapsam: governor deseni bir protokolü yönetir; STOA’nın primitif kütüphanesi genel olarak *kurumları* hedefler, söz dağarcığı yalnızca oylama ve hazine değil; vergi daireleri, tesciller, bütçeler, iş-emri masaları, tapu sicilleri ve tahkim içerir. İkincisi K3’ün tersine çevirmesi: governor dağıtımlarında hazine tipik olarak zaman kilidinin *mülküdür*, dolayısıyla zaman kilidinin yürüttüğü her şey harcayabilir; STOA’da ödenekleri *bütün* çağıranlara karşı hazinenin kendisi icra eder, “aşamaz”ı harfiyen kılan budur. Üçüncüsü, K1’in dürüstlük taksonomisi ile K2’nin kayıtlı kontrol grafinin, bildiğimiz kadarıyla karşılığı yoktur: mevcut çerçeveler hangi garantilerinin fiziksel, hangilerinin beyan olduğunu sistematik olarak ayırt etmez; dağıtıcının yetkisizleştirilmesini de manifestoya işlenen birinci sınıf bir dağıtım adımı olarak ele almaz. Protokol katmanında komşu mekanizmalar mevcuttur: Tezos, kendi yükseltme sürecine öz-değişikliği gömer; Aragon Court ve Kleros, teminatlı jürilerle uyuşmazlık çözer; MakerDAO’nun acil kapatması, protokole doğrudan bağlanmış bir son-çare sonucudur. Her biri kendi protokolünün parametrelerini yönetir; hiçbiri genel olarak kurumları hedeflemez ve hiçbirinde hazinenin kendisi operatörlerini reddetmez.

Topluluklar için yönetim araçları. PolicyKit, çevrimiçi toplulukların yönetim prosedürlerini platform API’lerine karşı çalışan kodla ifade etmesini sağlar [12]; daha geniş bir hareket, üyelerce yönetilen çevrimiçi mekânları inceler [13]. Bu sistemler değiştirilebilir platformları yönetir ve platformun iş birliğine dayanır; STOA ise icranın kendiliğinden yürümesi gereken, değer taşıyan ve çekişmeli uca odaklanır.

Kod-olarak-kurallar ve hesaplamalı hukuk. Kamu girişimleri mevzuatı makinece tüketilebilir biçimde yayımlamayı araştırır [14]; hesaplamalı hukuk yazını blokzincirin “lex cryptographica”sını inceler [5]. STOA bunların yapıcı tamamlayıcısı olarak okunabilir: mevcut hukuku koda çevirmek değil, işleyen kuralları baştan kod olan kurumların neye benzediğini sormak, ve bileşen başına, kodun bittiği ve beyanın başladığı yeri işaretlemek.

Kurumsal analiz. Primitif söz dağarcığımız, kurumsal iktisatçıların uzun süredir soyut düzlemde sorduğu bir soruya yazılımla verilmiş bir yanıttır: kurumlar neyden yapılmıştır? Ostrom’un

¹Artefakt deposu ve yeniden üretim talimatları gönderime eşlik edecektir; vaka çalışması, çerçevenin sözleşme test paketiyle deterministik olarak yeniden oynatılabilir.

müşterekler yönetişimi için tasarım ilkeleri, izleme, kademeli yaptırımlar, iç içe girişimler [15], ile Crawford–Ostrom kurum grameri [16], kurumları tipli ve birleştirilebilir ögelere ayrıştırır. STOA’nın kütüphanesi bilinçli olarak bunlarla uyumludur: birleştirilebilir sözleşmeler olarak gruplar, otoriteler, meclisler, yükümlülükler ve yaptırımlar; iç içe geçme olarak federasyon. Bu eşlemeye gelecek çalışmalarda dönüyoruz; $\diamond$ bileşenler olarak biçimselleştirdiğimiz oracle sınırı, bilinen blokzincir oracle problemini yankılar [17].

3 Çerçeve: Sözleşmeler ve Kablolar Olarak Kurumlar

3.1 Tek graf, üç okuma

Bir STOA sistemi bir graftır. Dügümler *çiplerdir*: çekirdek kütüphaneden primitif örnekleri; her birinin bir görünen etiketi, bir yapılandırması ve tipli *portları* vardır, çıkış portları sözleşme olaylarına, giriş portları sözleşme metotlarına karşılık gelir. Kablolar, bir olay portunu bir metot portuna, isteğe bağlı bir yük eşlemesiyle bağlar. Aynı graf üç biçimde okunur: tasarımcının düzenleyip canlandırdığı *süreç haritası*; *dağıtım manifestosu* (hangi sözleşmelerin hangi kurucu argümanlarla dağıtılacağı, ardından hangi bağlama çağrılarının yapılacağı); ve dağıtımdan sonra *canlı sistemin* adres defteri. Şablonlar, mevcut dağıtımda, bir köy federasyonundan ve bir satın alma sürecinden bir yazılım şirketine ve iki karşıt ulus-devlete, 23 adet, yalnızca kaydedilmiş graflardır; çerçevenin tasarım kuralı, çekirdeğin alan-nötr kalmasıdır: alan içeriği çekirdek kodda dallanma olarak değil, çıkarılabilir paketler olarak gelir.

3.2 Primitif kütüphanesi

Mevcut kütüphane, yaklaşık 30 Solidity sözleşmesiyle (2.808 satır) gerçekleştirilen 25 primitiften oluşur. Tablo 1, icra sınıflarıyla birlikte temsili bir alt kümeyi gösterir. Kütüphane bilinçli olarak küçüktür: kurumlar *kompozisyon* olarak ifade edilir; kompozisyonla ifade edilemeyen bir yetenek, bir sistemin içine gömülmenin değil, kütüphaneyi genişletmenin işaretidir.

3.3 Dürüstlük taksonomisi

Her primitif bir **enforcement** alanı taşır: *öz-icralı* ya da *beyana dayalı*. Alan, dokümantasyon değildir; tasarım aracının teknik görünümünde bir rozet olarak çizilir ve dağıtım manifestosuna yazılır, ayırım, tasarımdan operasyona dek yaşar. Mesele ürün bütünlüğüdür: gerçek bir organizasyon için *hangi kuralların çiğnenebileceğini* bilmek, ürünün ta kendisidir.

İki sözleşme, iki yarıyı birden içerdiği için sınırı keskin biçimde gösterir. İtfa masası, itfa talebi anında şirket parasını (scrip) yakar, arz geri dönülmez biçimde daralır; öz-icralı bir adım, karşılık gelen banka havalesi ise bir maliye görevlisinin sonradan ödendi olarak işaretlediği, izlenen ve *beyana dayalı* bir adımdır. Fatura defteri, zincir-üstü ödemeleri doğrudan hazineye işler (öz-icralı); banka yoluyla gelen ödemeler içinse görevlinin beyanını kabul eder (beyana dayalı). Taksonomi, tasarımcıyı kurumunun tam olarak nerede fizik olmaktan çıkıp güvene dönüştüğünü görmeye zorlar; oracle’a bağımlı yüzeyi örtük olmaktan çıkarıp açık ve sayılabilir kıldığı için de standart oracle itirazını [17] baştan soğurur.

Primitif (sözleşme)	Kurumsal rol	Sınıf
Sicil (StoaGroup)	üyelik, roller, uygunluk	◆
Meclis (StoaAssembly)	teklifler, nisap, kademeli oylar	◆
Zaman kilidi (StoaTimeLock)	itiraz penceresi, askıya alma/sürdürme	◆
Hazine (StoaAccount)	bakiyeler, korunum, harcama politikası	◆
Bütçe (StoaBudget)	kategori/dönem başına ödenekler	◆
Para birimi (StoaToken)	ERC-20 sistem parası, basım politikası	◆
Akış (StoaFlow)	hesaplar arası değer hareketi	◆
Tescil sicili (StoaCharterRegistry)	izinsiz tüzel-kışı kaydı	◆
Vergi dairesi (StoaTaxCollector)	sicile bağlı tahsilat	◆
İş-emri masası (StoaWorkOrder)	ödül emaneti, süre sonu iadesi	◆
Fatura defteri (StoaInvoice)	alacaklar; zincir-üstü tahsilat	karma
İtfa masası (StoaRedemptionDesk)	şirket parasından çıkış; banka bacağı	karma
Tahkim (StoaArbitration)	davalar ve hükümler	◇
Oracle (StoaOracle)	dış olgular, tek raportör	◇
Kayıt (StoaRecord)	sertifika, siciller	◇

Tablo 1: Temsilî primitifler. ◆ = öz-icralı (kural, sözleşmenin denetlediği durum üzerinde zincir üstünde sınanır; ihlal geri döner). ◇ = beyana dayalı (zincir dışı gerçekliğe ilişkin noterlenmiş iddia). “Karma” sözleşmeler iki yarıyı da açıkça içerir. Genel: 25 primitifin 16’sı öz-icralı, 9’u beyana dayalı.

4 Aktüasyon Döngüsü

4.1 Yürütülebilir teklifler

Meclis iki tür teklif kabul eder. Metin teklifi bir başlık ile gövdedir, noterlenmiş bir niyet. *Eylemli* bir teklif, ek olarak paralel hedef ve çağrı-verisi dizileri taşır:

```
createProposalWithActions(title, body, votingSeconds,
                          targets, calldatas, category)
```

Oylama bir-adres-bir-oydur; takılabilir bir üyelik sicili (`isMember(address)` sunan herhangi bir sözleşme) ile kapılanır; oy pusulaları evet/hayır/çekimser olup oylama pencereleri gerçek zamanlıdır. Sonuçlandırma sırasıyla şunları sınar: bir nisap kuralı (yapılandırılmış n/d için $kullanılan \cdot d \geq seçmen \cdot n$), ardından evet+hayır oyları üzerinden baz puan cinsinden bir onay eşiği. Eşik *kategoriye göre kademelidir*: **existential** gibi bir kategori 7.500 baz puana bağlanabilir; böylece sıradan bir kararı geçiren %60’lık çoğunluk, bir feshi geçiremez, tek bir eşlemede ifade edilmiş anayasal katılık. Onay üzerine meclis, teklifin eylemlerini bağlı zaman kilidine kuyruklar ve **ProposalQueued** yayar; metin teklifleri hiçbir şey kuyruklamaz. Karar artık rayındadır: bundan sonrası, herhangi bir operatörün uymayı seçmesine bağlı değildir.

4.2 İtiraz penceresinin dışları var

Zaman kilidi, kuyruklanan işlemleri yapılandırılmış gerçek-zamanlı bir gecikme, bir *itiraz penceresi*, boyunca tutar; sonrasında bunları *herhangi biri* yürütebilir; yürütme, çağrıları yapar ve içteki geri dönüş gerekçelerini yüzeye taşır. Pencere açıkken işlem, yönetici ya da teklif sahibi tarafından iptal edilebilir; bağlanmış bir *hakem* (bir mahkeme çipi) ise işlemi **suspend** ile askıya alabilir, askıdaki bir işlem, penceresi dolsa bile yürütülemez, sonra hükümle **resume** eder ya da

iptal eder. Bu, bilinçli olarak bir yargının tohumudur: dava açmak yürütmeyi dondurur; hüküm, edimi ya serbest bırakır ya da hükümsüz kılar (§8).

4.3 Yönetişim devri: sistem kendi üzerinde egemenleşir

Her primitif, yöneticisi dağıtıcısı olarak konuşlanır, kuruluş anında bundan kaçış yoktur; ve bu, dağıtılmış yönetimdeki en derin bütünlük kusuru saydığımız şeyin köküdür: hesap vermeyen süper-kullanıcı. Anayasası, dağıtıcısınca aşılabilen bir mekanizma bir demodur. STOA bu yüzden *yetkisizleştirmeyi bir dağıtım adımı* olarak ele alır: bağlama tamamlandıktan sonra hat, her çipin yönetim haklarını atanmış bir yöneten çipe, referans yapılandırma, tek teklif sahibi meclis olan zaman kilidine, devreder ve ortaya çıkan *kontrol grafını* (kimin neyi yönettiğini) kullanıcının gördüğü manifestoya işler. O işleminden itibaren bütçeyi değiştirmek, zaman kilidini yeniden bağlamak ya da hazineye operatör atamak, yalnızca geçmiş bir oylamanın çıktısı olarak mümkündür. Dağıtıcının elinde, herhangi bir üyenin haklarının tıpkısı kalır: teklif etmek ve oy vermek.

5 Kod-Olarak-Bütçe

Hazine primitifi, iki çıkış metodu (bir adrese çekim; eş bir hesaba aktarım), bir operatör kümesi ve isteğe bağlı bir *harcama politikası* ile korunumlu bir bakiye tutar. Politika atandığında, *her* çıkış yolu önce politika sözleşmesinde

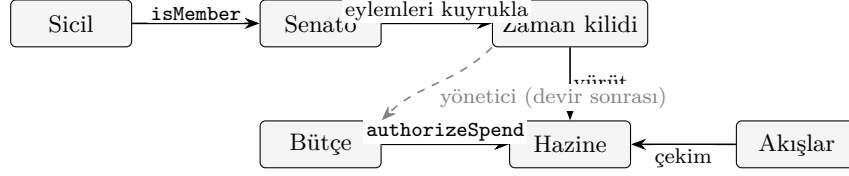
```
authorizeSpend(operator, to, amount)
```

çağrısını yapar; reddetmek için politika geri dönmek (revert) zorundadır, ve bu sına, bakiye muhasebesinden önce ve çağırana bakılmaksızın, *hazinenin kendi sahibi dahil*, çalışır. Bütçe primitifi bu arabirimi bir ödenek defteri olarak gerçekler: kategori ve mali dönem başına adlandırılmış kalemler; harcamacılar (akış çipleri, görevliler) kategorilere atanır; kalemi olmayan bir harcama No **appropriation line**, kalem bakiyesini aşan bir harcama **Appropriation exceeded** gerekçesiyle geri döner; yetkilenen her harcama kalemi atomik olarak artırır. Mali dönemler açıktır: yeni dönem *boş* başlar; harcama yetkisi, yeniden ödenekleştirilene dek kendiliğinden düşer, bir bütçe kanununun süresinin dolmasının zincir-üstü karşılığı.

İki tasarım tercihi önemlidir. Birincisi, icra noktası yönetim yığını değil hazinedir: standart governor düzeni *yönetişimin* yürüttüklerini kapılar, ama hazineyi operatörlerine itaatkâr bırakır; sınaıyı hazinenin çıkış yoluna yerleştirmek, operatör/sahip bypass'ını kapatır, gerçek kurumların kapatamadığı delik tam budur (imza yetkili bir görevli kâğıt bütçeyi *aşabilir*; burada işlem geri döner). İkincisi, bütçenin yöneticisinin zaman kilidi olması amaçlanır (§4.3); böylece ödenek kalemlerinin kendisi de yalnızca geçmiş oyla değişir: meclis, sözcüğün tam anlamıyla bütçe komisyonudur ve onayladığı bütçe bir talimat değil, harcamanın fiziksel dış sınırıdır. Hazine, simülasyon için sembolik-defter modunda; sistemin ERC-20 para birimine bağlandığında ise aynı politikanın gerçek token hareketini kapıladığı saklama (custody) modunda çalışır, bütçe sınaı, tek bir token çıkmadan önce koşar.

6 Vaka Çalışması: Bütçesi Aşlamayan Asgari Bir Devlet

Beş çip birleştiriyoruz: bir hazine (1.000 birim), ona harcama politikası olarak bağlanmış bir bütçe, gerçek-zamanlı itiraz pencereci bir zaman kilidi, bir senato (nisap 1/2, olağan eşik 5.000



Şekil 1: Vaka çalışması kompozisyonu. Senato, onaylanan eylemleri zaman kilidine kuyruklar; itiraz penceresi dolunca bunları herhangi biri hazineye karşı yürütebilir; hazine, akışlar ve kendi sahibi dahil her çıkışta bütçeye danışır. Devirden sonra (kesikli) bütçe, hazine ve senato üzerindeki yönetici koltukları, tek teklif sahibi senato olan zaman kilidindedir.

Girişim	Sonuç
Dağıtıcı, bütçede doğrudan ödenek tanımlar	revert: Only admin
Dağıtıcı, hazineye kendine operatör yetkisi verir	revert: Only owner
Dağıtıcı, senatoyu yeni bir zaman kilidine yönlendirir	revert: Only admin
Atanmamış akış ödeme tetikler (bütçe kalemi yok)	revert: No appropriation line
Hazine sahibi kalemsiz harcar (“owner grab”)	revert: No appropriation line
Atanmış akış, 100 birimlik kalemini 1 birim aşar	revert: Appropriation exceeded
Kuyruktaki işlem, pencere dolmadan yürütülür	revert: Appeal window open

Tablo 2: Yönetişim devrinden sonra vaka sistemi üzerindeki bypass girişimleri. Harcamaya ya da kural değişikliğine giden anayasa-dışı her yol geri döner.

baz puan) ve ödeme akışları (Şekil 1).

Dağıtım hattı, §4.3’deki devri uygular: bütçe yöneticisi → zaman kilidi, hazine sahibi → zaman kilidi, senato yöneticisi → zaman kilidi; zaman kilidinin tek teklif sahibi senatodur. Senaryo, çerçevenin sözleşme test paketinde deterministik olarak yeniden oynar; aynı kompozisyon bir şablon olarak dağıtılmakta olup çerçevenin çalışma zamanında canlı bir örnek olarak da işletilmiştir.

6.1 Her bypass girişimi geri döner

Tablo 2 girişimleri listeler. Dağıtıcı, sistemdeki her sözleşmeyi yaratan hesap, artık ödenek tanımlayamaz, operatör hakkı kapamaz, yönetişimi yeniden bağlayamaz; bütçesiz harcamacılar tek bir birim kıpırdatamaz; bütçeli harcamacılar kalemlerini tek bir birim aşamaz.

6.2 Anayasal yol başarıya ulaşır

Ardından bir üye, dağıtıcının yapamadığını hukuk içinde yapar: bütçe çipini hedefleyen `setAppropriation("schools", 500)` eylemini taşıyan “FY-1 okullar bütçesi” teklifini verir; üyeler oy kullanır; teklif nisabı ve eşiği karşılar; sonuçlandırma eylemi zaman kilidine kuyruklar; itiraz penceresi askıya alınmadan dolar; işlemi *herhangi biri* yürütür. Ödenek kalemi artık 500 birimle yaşamaktadır; atanmış akışlarınca tükenene ya da dönem devrilene kadar harcanabilir. Tek bir kompozisyon içinde, işleyen tek yol oydur, ve oy, oylanmış eylemi *kanıtlanabilir biçimde doğrurur*: onay kuyruklar, pencere korur, yürütme izinsizdir; sonrasındaki icra, hazinenin kendi ret mantığıdır.

§3.3 taksonomisi altında bu döngüdeki her şey ♦’dir: teklif kabulü, pusulalar, nisap aritmetiği, kademeli eşikler, kuyruklama, pencere, yürütme, ödenek muhasebesi ve harcama kapısı. ♦

kalan ise tam olarak kalması gereken şeydir: *okulların gerçekten yapıp yapılmadığı* dünyaya dair bir iddiadır ve sistem bunu ancak noterleyebilir, tasarım yüzeyinin gizlemek yerine gösterdiği bir sınırdır.

7 Kodlanamayanlar

Her garantiyi alt katman sınırlar. Yukarıdaki bütün iddialar zincire görelidir: tek-operatörlü bir zincirde “sözleşme X’i garanti eder”, “operatör X’e söz verir” demektir. Çerçevenin yerel çalışma zamanı bu yüzden *simülasyon modu* olarak konumlandırılır, tasarım, prova ve araştırma için uygundur, ve garantiler ancak katılımcıların tek tek denetlemediği bir alt katmanda (kamusal bir ağ ya da her federe organizasyonun doğrulayıcı olduğu bir konsorsiyum zinciri) çekişmeli anlamda anlam kazanır. Bu, çerçevenin gömmesi değil, yüzeye çıkarması gereken bir dağıtım kararıdır.

Bir anahtar çifti bir kişi değildir. Bir-adres-bir-oy, kimliği varsayar. Anahtar kaybı, anahtar hırsızlığı, vasilik, ölümden veraset ve Sybil direnci mevcut kütüphanede çözülmemiştir; bunları salt zincir-üstü hiçbir kurgu da çözmez: kurtarma ve tasdik içeren bir kimlik katmanı gerekir, ki bu, güvenilen taraflar ithal eder ve defterin $\diamond$ tarafına aittir. İlgili olarak, meclisin seçmen-nüfus parametresi bugün yönetsel olarak (devirden sonra: oyla) belirlenir; sicile bağlanması mekaniktir, ama *kimin üye sayılacağı* kodun yanıtlayabileceği bir soru değildir.

Oylar açıktır. Pusulalar zincir üzerinde sonsuza dek okunabilir. Baskıya ve oy satın almaya direnç gizli oy gerektirir (asgari olarak commit-reveal); gizli kurumsal kayıtlar ise kamusal durumla çelişir. Bunu, buradaki “seçim” sözcüğünün anlamından bilinen bir eksilme olarak işaretliyoruz.

Hukuki tanınma yoktur. Geçen bir teklif çipleri bağlar; bir mahkemeyi, bir bankayı ya da bir vergi dairesini bağlamaz. Hukuki gerçekliğe köprü, aynı graftan üretilen yargı-alanına özgü sarmalayıcılar (dernek tüzükleri, şirket ana sözleşmeleri) gerektirir, gerçek benimseme için isteğe bağlı değil, zorunlu saydığımız bir gelecek çalışması.

8 Gelecek Çalışmalar

Tipli yamalar olarak anayasa değişikliği. Kurumların en çok yük taşıyan kuralı, kuralları değiştirme kuralıdır. Değişim birimi olarak *yamalar* geliştiriyoruz: örneklere değil, primitif söz dağarcığına karşı tiplenmiş alt-graf farkları (“**treasury** türünde bir bağlantı noktası gerektirir”). Canlı bir sistem yamayı oyla benimser; yama uygulamak *anayasa değişikliğinin ta kendisidir* ve aynı biçim, iyileştirmeleri söz dağarcığı üzerine kurulu bütün sistemlere dağıtır. Doğrulanmamış bir yama bir anayasal darbe vektörüdür; bu da doğrulamayı (aşağıda) yük taşıyıcı kılar.

Hüküm, infazdır. Zaman kilidinin askıya alma/sürdürme/iptal hakem kancaları, hüküm ile icramın tek bir atomik işlemde çakıştığı bir yargının tohumudur, tazminat aktarılır, ruhsat iptal edilir, kayıt hükümsüz kılınır; hepsi hüküm işleminin içinde, yargı yetkisi, tescil anında rızayla verilmiş olmak kaydıyla. Bunun ima ettiği güç yoğunlaşması (mahkemenin ele geçirilmesi), birinci sınıf desenler olarak tasarlamayı planladığımız karşı-ağırlıkları zorunlu kılar: yargıç görev süreleri, temyiz katmanları, yargı yetkisinin kapsamlanması.

Kurumsal tasarımların doğrulanması. Çerçeve, primitif söz dağarcığı üzerinde graf öne-
ren LLM destekli bir tasarımcı içerir. Üretilmiş anayasalar, yapısal doğrulamadan fazlasını hak
eder: dağıtım öncesinde ve yama-benimseme anında koşan, kurumsal değişmezleri, nisabın eri-
şilebilirliği, (dağıtıcı dahil) tek-başına-boşaltıcı yokluğu, her yetki için bir itiraz yolu, sahipsiz
yönetici yokluğu, kademeli yaptırımlar [15], sınavan bir özellik katmanı planlıyoruz. Kütüpha-
nenin boyutu (2.808 satır), primitiflerin kendilerinin biçimsel doğrulamasını da erişilebilir kılar.

Seçimler, makamlar ve kurumsal omurga. Temsilî yapılar (adaylık, görev süreleri, geri
çağırma), temerrüt sonuçlu yükümlülükler, pay defteri üzerinden ağırlıklı oy ve gizli oy sıradaki
primitiflerdir; her biri söz dağarcığı için bir kompozisyon testidir.

9 Sonuç

Bir kararı kaydetmek ile ona sebep olmak arasındaki mesafe, bir noter ile bir mekanizma arasın-
daki mesafedir. Anlamli bir kurumsal kural sınıfı için bu mesafenin kapandığı, küçük ve alan-nötr
bir primitif kütüphanesi gösterdik: oylanın eylemi kanıtlanabilir biçimde doğuran bir oy, yarat-
tığı sistemi kanıtlanabilir biçimde aşamayan bir dağıtıcı ve kanıtlanabilir biçimde aşlamayan bir
bütçe, icranın bittiği ve beyanın başladığı yeri işaretleyen, makinece taşınan dürüst bir sınırla
birlikte. Kendimize koyduğumuz ve alana önerdiğimiz çıta, yukarıdakilerin tümünü değiştirme
kurallarının da zincir üstünde olmasıdır; oyla benimsenen tipli yamalar, ona doğru attığımız bir
sonraki adımdır.

Kaynakça

- [1] Vitalik Buterin. Ethereum: A next-generation smart contract and decentralized application platform. <https://ethereum.org/en/whitepaper/>, 2014. Whitepaper.
- [2] Gavin Wood. Ethereum: A secure decentralised generalised transaction ledger. <https://ethereum.github.io/yellowpaper/paper.pdf>, 2014. Yellow paper.
- [3] Lawrence Lessig. *Code and Other Laws of Cyberspace*. Basic Books, New York, 1999.
- [4] Nick Szabo. Formalizing and securing relationships on public networks. *First Monday*, 2(9), 1997.
- [5] Primavera De Filippi and Aaron Wright. *Blockchain and the Law: The Rule of Code*. Harvard University Press, Cambridge, MA, 2018.
- [6] OpenZeppelin. Governance — OpenZeppelin contracts documentation. <https://docs.openzeppelin.com/contracts/governance>. Accessed July 2026.
- [7] Luis Cuende and Jorge Izquierdo. Aragon network: A decentralized infrastructure for value exchange. <https://github.com/aragon/whitepaper>, 2017. Whitepaper.
- [8] Shuai Wang, Wenwen Ding, Juanjuan Li, Yong Yuan, Liwei Ouyang, and Fei-Yue Wang. Decentralized autonomous organizations: Concept, model, and applications. *IEEE Transactions on Computational Social Systems*, 6(5):870–878, 2019.

- [9] Samer Hassan and Primavera De Filippi. Decentralized autonomous organization. *Internet Policy Review*, 10(2), 2021.
- [10] Youssef El Faqir, Javier Arroyo, and Samer Hassan. An overview of decentralized autonomous organizations on the blockchain. In *Proceedings of the 16th International Symposium on Open Collaboration (OpenSym 2020)*. ACM, 2020.
- [11] Vitalik Buterin. DAOs, DACs, DAs and more: An incomplete terminology guide. <https://blog.ethereum.org/2014/05/06/daos-dacs-das-and-more-an-incomplete-terminology-guide>, 2014. Ethereum Foundation Blog.
- [12] Amy X. Zhang, Grant Hugh, and Michael S. Bernstein. PolicyKit: Building governance in online communities. In *Proceedings of the 33rd Annual ACM Symposium on User Interface Software and Technology (UIST '20)*, pages 365–378. ACM, 2020.
- [13] Nathan Schneider. *Governable Spaces: Democratic Design for Online Life*. University of California Press, Oakland, 2024.
- [14] James Mohun and Alex Roberts. Cracking the code: Rulemaking for humans and machines. OECD Working Papers on Public Governance 42, OECD, 2020.
- [15] Elinor Ostrom. *Governing the Commons: The Evolution of Institutions for Collective Action*. Cambridge University Press, Cambridge, 1990.
- [16] Sue E. S. Crawford and Elinor Ostrom. A grammar of institutions. *American Political Science Review*, 89(3):582–600, 1995.
- [17] Giulio Caldarelli. Understanding the blockchain oracle problem: A call for action. *Information*, 11(11):509, 2020.