TAMGA NETWORK

TAMGA NETWORK AKADEMIK ÇALIŞMALAR

İmzalı Listelerden Deftere

Tamga Network'ün Zincir Tasarımı

Tamga Network

KASIM 2026 · SÜRÜM 1.0

© 2026 Tamga Network. Bu eser Creative Commons Atıf 4.0 Uluslararası Lisansı (CC BY 4.0) ile yayımlanmıştır: kaynak gösterilerek paylaşılabilir ve uyarlanabilir. https://creativecommons.org/licenses/by/4.0/deed.tr

Özet

Tamga Network, Türkiye ve Türk dünyası için tasarlanmış, Avrupa Birliği'nin dijital kimlik düzeniyle aynı teknik dili konuşan bir güven ağıdır. Ağın güveni bugün imzalı ve sürümlü güven listelerinde ve saatlik bir çapa günlüğünde durur; çalışan bir zincir yoktur. Ağın ilk tasarımı bir blokzincirle başlıyordu; bu yazı o zincirin neden ertelendiğini, ertelenirken neden bir kenara atılmadığını ve bugün hangi kurallarla bekletildiğini anlatıyor. Merkezdeki fikir projeksiyon ilkesidir: defter açıldığında da kaynak imzalı listeler olarak kalır, defter yalnız onların ikinci ve değiştirilemez bir kaydı olur. Doğrulama hiçbir adımda deftere bağlı değildir ve deftere hiçbir zaman kişisel veri yazılmaz.

Yazı, zincirsiz hâlin nasıl çalıştığından başlayıp Hyperledger Besu ve QBFT üzerine kurulan izinli defteri, kontratların haritasını, listelerdeki her güven kaydının deftere nasıl ve hangi sınırla yansıdığını, belgenin verildiği andaki durumun değerlendirilmesini, sert iptali, doğrulayıcı kimliğinin alan adına bağlanmasını ve devrini adım adım açıklıyor. Ardından listelerin geçmişinin deftere bir kez ve denetimli bir pencerede aktarılmasını, bu aktarımın doğruluğunu sınayan eşdeğerlik testinin yöntemini ve yönetişim kurallarını anlatıyor. Her parçanın bugünkü durumu "Çalışıyor", "Planlanan" ya da "Araştırma" etiketiyle verilmiştir. Kısa cevap şudur: defterin kodu yazılmış ve sınanmıştır, üretimde kapalıdır ve ikinci bağımsız bir işletmeci ile bağımsız bir güvenlik denetimi olmadan açılmayacaktır.

Anahtar kelimeler: güven listeleri · çapa günlüğü · izinli defter · Hyperledger Besu · QBFT · projeksiyon · eşikli çapa · yeniden oynatma · eşdeğerlik testi · göç penceresi · sert iptal · yönetişim

ATIF

Tamga Network (2026). İmzalı Listelerden Deftere: Tamga Network'ün Zincir Tasarımı. Sürüm 1.0. https://tamga.network/files/publications/signed-lists-to-ledger-tr.pdf

İçindekiler

1 Giriş: Bir Diploma, İki Soru 3
2 Önce İmzalı Listeler: Zincirsiz Hâl 3
2.1 Üç dosya 3
2.2 Sürüm zinciri ve saatlik günlük 4
2.3 Doğrulama listeye sorar, kimseye sormaz 5
2.4 Neden önce listeler? 5
3 Zincir Neyi Ekler, Neyi Eklemez? 6
3.1 Çift konuşma 6
3.2 Defterin ekleyebileceği iki şey 6
3.3 Defterin eklemeyeceği şeyler 7
4 Defterin Motoru: Besu, QBFT ve İzinli Ağ 7
4.1 Neden Besu ve QBFT? 7
4.2 İzinli ağ: kim bağlanır, kim yazar? 8
5 Kontratların Haritası 9
5.1 Dokuz kontrat ve bir okuma kapısı 9
5.2 Egemenlik kodda 10
5.3 Doğrulamanın beş sorusu 10
6 Projeksiyon İlkesi ve Liste Çapası 11
6.1 Kaynak liste, türev defter 11
6.2 Zincire yazılan iki kayıt 11
6.3 Eşikli yazıcılar ve rakip adaylar 12
6.4 Doğrulayıcının üç cevabı 13
7 “Her Şey Zincirde”: Kapsam ve Kişisel Veri Sınırı 14
7.1 Liste özetinden kayıt düzeyine 14
7.2 Sınır: özet yazılır, metin yazılmaz 14
7.3 Ağ düzeyi kayıtlar ve yeni günlük satırları 15
8 Belgenin Verildiği Andaki Durum 16
8.1 Kapanan kurumun eski diploması 16
8.2 Durum adlarının eşlenmesi 17
8.3 Sert iptal: “şu andan sonrasını sayma” 17
9 Doğrulayıcı Kimliği: Alan Adı, Devir ve Yeniden Kayıt 18
9.1 Kimlik sertifikadan alan adına 18
9.2 İki sorun: kapma ve kalıcı kilit 18
9.3 Çözüm: dönemler ve ortak oy 19
10 Listelerden Deftere Geçiş 19
10.1 Geçiş bir yeniden yazım değil 19
10.2 Bugünkü kontratlar geçmişi neden yazamıyordu? 19
10.3 Tek seferlik göç penceresi 20
10.4 Eşdeğerlik nasıl sınanır? 21
11 Yönetişim 22
11.1 Üç katman 22
11.2 Yükseltmede bekleme süresi 23
12 Bugün Nerede? 23
13 Sınırlar ve Açık Sorular 24
14 Sonuç 25
Kaynaklar 25
Ek A — Sözlük 27

1 Giriş: Bir Diploma, İki Soru

Aynur yüksek lisansını İstanbul'da bitirdi. Diploması, üniversitenin imzaladığı dijital bir belge olarak telefonundaki cüzdanda duruyor. Bir gün bir işverene başvuruyor; işveren belgeyi istiyor, Aynur yalnız istenen alanları gösteriyor ve birkaç saniye sonra ekranda "doğrulandı" yazıyor. Bu birkaç saniyede işverenin yazılımı üç şeyi denetledi: imza gerçekten o üniversitenin anahtarından mı geliyor, üniversite o tarihte diploma vermeye yetkili miydi, belge sonradan iptal edildi mi. İlk iki sorunun cevabını işverene bir **güven listesi** verdi.

Aynı gün, başka bir şehirde Timur adında bir bilgi işlem sorumlusu, çalıştığı kurumun bu ağda yalnız belge veren olarak değil, ağın kayıtlarını birlikte tutan ikinci bir taraf olarak da yer alıp alamayacağını düşünüyor. Timur'un önündeki belgelerde iki soru var. Birincisi: Bugün bu listeleri tek bir işletmeci imzalıyorsa, o işletmecinin iki farklı doğrulayıcıya iki farklı liste göstermediğinden nasıl emin olabiliriz? İkincisi: Ağın ilk tasarımında bir blokzincir vardı; şimdi neden yok, gelecekse ne zaman ve hangi kurallarla gelecek?

Bu yazı Timur'un iki sorusuna cevap veriyor ve bunu yaparken Aynur'un diplomasını hiç gözden kaçırmıyor. Çünkü bir güven ağında altyapı ne kadar değişirse değişsin, ölçü aynı kalır: Aynur'un bugün geçerli olan diploması, defter açıldıktan sonra da aynı cevabı almalıdır. Zincir tasarımındaki kararların neredeyse hepsi bu tek ölçünün sonucudur.

Blokzincirin kimlik için genel olarak ne zaman işe yaradığı sorusunu dizinin on ikinci yazısı [2] kavramsal düzeyde tartıştı. Orada varılan sonuç, zincirin güvenin kaynağı olamayacağı, ama imzalı kayıtların sessizce değiştirilmediğine birden çok bağımsız tarafın tanıklık etmesini sağlayabileceğiydi. Bu yazı o genel tartışmayı tekrar etmiyor. Burada anlatılan, o fikrin Tamga Network'te nasıl somut bir tasarıma dönüştüğüdür: hangi dosya, hangi kontrat, hangi alan, hangi kural.

Yazının planı şöyle. Bölüm 2 zincirsiz hâli, yani bugün çalışan imzalı listeleri ve çapa günlüğünü anlatıyor. Bölüm 3 bir defterin bu düzene neyi ekleyip neyi eklemeyeceğini ayırıyor. Bölüm 4 defterin motorunu (Hyperledger Besu ve QBFT) ve izinli ağın kurallarını, Bölüm 5 kontratların haritasını veriyor. Bölüm 6 projeksiyon ilkesini ve liste çapalarını, Bölüm 7 "her şey zincirde" kapsamını ve kişisel veri sınırını inceliyor. Bölüm 8 belgenin verildiği andaki durumu ve sert iptali, Bölüm 9 doğrulayıcı kimliğini ve devrini ele alıyor. Bölüm 10 listelerden deftere geçişi, yani yeniden oynatmayı, göç penceresini ve eşdeğerlik testini anlatıyor. Bölüm 11 yönetişimi, Bölüm 12 bugünkü durumu, Bölüm 13 sınırları ve açık soruları kapsıyor.

> ÖNEMLİ
>
> **Durum etiketleri**
>
> Bu yazıda her parçanın yanında üç etiketten biri bulunur. **Çalışıyor**: kodu yazılmış ve testleri geçiyor; imzalı listeler ve çapa günlüğü için bu, gerçek ağda canlı olmak demektir, defter parçaları için yalnız yerel geliştirme ağında çalışmak demektir. **Planlanan**: kuralı karara bağlanmış, ama bir önkoşul (ikinci işletmeci, bağımsız denetim, yönetişim kurumu) beklendiği için üretimde yok. **Araştırma**: bağlayıcı olmayan, yalnız kayıtta duran bir yön. Hiçbir yerde "çalışıyor" ifadesi "üretimde açık bir zincir var" anlamına gelmez; üretimde açık bir zincir yoktur.

2 Önce İmzalı Listeler: Zincirsiz Hâl

2.1 Üç dosya

Tamga Network'ün güveni bugün herkese açık bir adreste (`trust.tamga.network`) yayımlanan birkaç dosyada durur [13]. En üstte **listelerin listesi** (`lotl.jws`) vardır. Bu dosya ağdaki ülke listelerinin adreslerini ve onları imzalayan anahtarları, ağ düzeyindeki belge türlerini, cüzdan sağlayıcılarını ve

ortak ayarları sayar. Altında **ülke listeleri** bulunur; bugün etkin olan tek liste Türkiye listesidir (`tl-tr.jws`) ve kök sertifikaları, belge veren kurumları, kayıtlı doğrulayıcıları ve ülkeye özgü belge türlerini içerir. Üçüncü dosya **çapa günlüğüdür** (`anchors.jsonl`): iptal listelerinin her yayınını ve belge türü tanımlarındaki her değişikliği zaman sırasıyla kaydeden, her satırı ayrı imzalı bir günlük.

Bu yapı bilinçli bir seçimdir. Avrupa Birliği'nin dijital kimlik cüzdanı ekosisteminde de güvenin dayanağı her üye devletin imzaladığı güven listeleri ve Komisyon'un bunları toplayan listelerin listesidir; listelerin biçimi ETSI TS 119 612'de tanımlıdır [16]. Tamga'nın listeleri aynı mantığı izler ve ülke listesi ETSI'nin XML biçiminde ikinci bir görünüm olarak da üretilir.

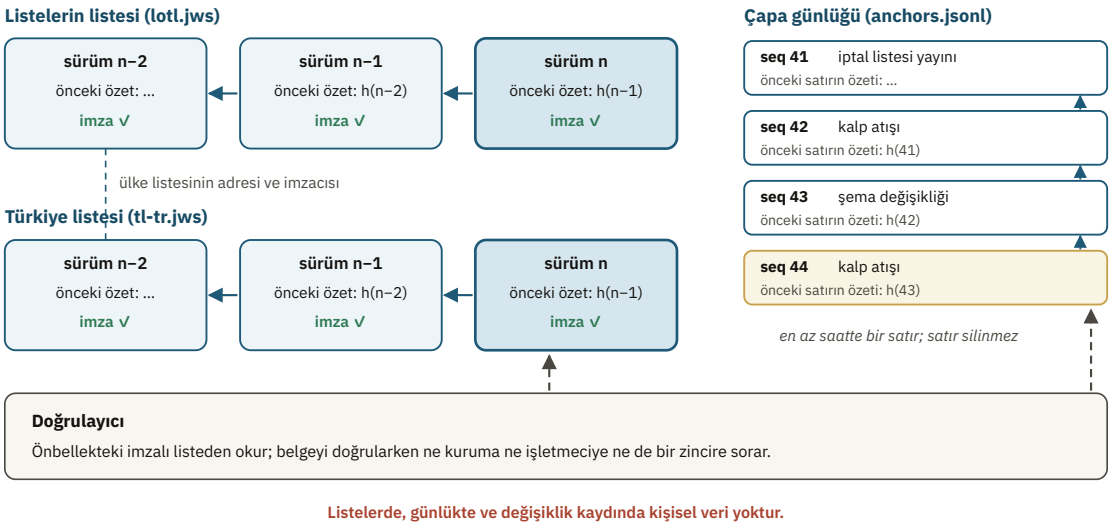
Listelerde kişisel veri yoktur. Bir kurum kaydında kurumun adı, sertifikasının parmak izi, durumu, tarihler ve adresler bulunur; bir kişinin adı, kimlik numarası ya da bir belgenin özeti hiçbir listede, günlükte ya da değişiklik kaydında yer almaz (SPEC-TRUST-0001 TL11) [13].

2.2 Sürüm zinciri ve saatlik günlük

Listelerin güvenilir olmasını sağlayan iki kural vardır. Birincisi **sürüm zinciridir**. Her liste bir sürüm numarası taşır ve bu numara yalnız artar. Her yeni sürüm bir öncekinin SHA-256 özetini içinde taşır (`previous_version_hash`). Bu sayede bir doğrulayıcıya eski bir sürümü yeni diye sunmak ya da geçmişten bir sürümü sessizce değiştirmek fark edilir: zincirin bir halkası değişirse ondan sonraki halkaların hepsi tutmaz hâle gelir. Bu fikir yeni değildir; belgeleri birbirine özetlerle bağlayarak zaman damgalamak 1991'de önerilmişti [17].

> TEKNİK AYRINTI
>
> **Sürüm zincirinin kuralı**
>
> L_n bir listenin n'inci sürümünün imzalı hâli, prev_n de bu sürümün içinde taşınan önceki sürüm özeti olsun (SPEC-TRUST-0001 §1) [13]:
>
> $$\begin{aligned} \mathrm{prev}_n &= \mathrm{SHA\text{-}256}(L_{n-1}) \\ \text{sürüm}(L_n) &> \text{sürüm}(L_{n-1}) \\ \mathrm{Verify}(L_n) = \text{geçerli} \quad &\Longleftrightarrow \quad \text{imza} \wedge \text{zincir} \wedge t < \text{next_update} \end{aligned}$$
>
> Son satırdaki tazelik koşulu şunu söyler: listenin bir sonraki güncelleme tarihi (`next_update`) geçmişse ya da listeye erişilemiyorsa doğrulama ne kabul ne ret üretir, "belirsiz" (INDETERMINATE) üretir (TL5). Listeler değişiklik olmasa da en geç 90 günde bir yeniden imzalanır.

İkinci kural **çapa günlüğüdür**. Günlüğün her satırı da bir öncekinin özetini taşır ve en az saatte bir satır yazılır; değişiklik yoksa bile "kalp atışı" (heartbeat) satırı eklenir. Böylece günlük, ağın saat saat ilerleyen bir tutanağına dönüşür. Satır silinmez; günlük büyüdükçe eski satırlar imzalı bir kontrol noktasıyla arşive taşınır ve arşiv herkese açık kalır (TL12) [13].



Şekil 1. Zincirsiz hâl: imzalı listeler, sürüm zinciri ve saatlik çapa günlüğü

2.3 Doğrulama listeye sorar, kimseye sormaz

Aynur'un diplomasını denetleyen işveren, listeyi önceden indirmiş ve imzasını denetlemiştir. Belgeyi doğrularken ne üniversiteye ne de ağın işletmecisine bir istek gönderir; gerekli bilgiyi önbellekteki imzalı listeden okur. Listeler `TrustSource` adlı tek bir arayüzün arkasındadır ve servislerin, doğrulayıcıların ve cüzdanın güven sorularını sorduğu yer burasıdır [15]. Liste dosyasını bu arayüzün dışında yorumlayan bileşen yoktur (TL4).

Bu düzenin bir yan sonucu önemlidir: işletmecinin yayın bileşeni bir süre çalışmasa bile doğrulama durmaz. Doğrulayıcı elindeki listenin süresi dolana kadar önbellekten doğrulamayı sürdürür; süre dolunca sonuç "belirsiz" olur [3]. Zincir tasarımında sık sık dönülecek bir ilke burada doğar: **doğrulama, kimseye sormadan yapılabilmelidir.**

2.4 Neden önce listeler?

Tamga'nın ilk yol haritası pilotu bir zincire bağlıyordu: kontratlar derlenip testnet ayağa kalkmadan gerçek bir kurum ağa katılamayacaktı. ADR-0009 bu sırayı tersine çevirdi [3]. Gerekçesi basit bir soruydu. İlk aşamada dört blok onaylayıcısı (validator) planlanmıştı ve dördü de aynı işletmecinin olacaktı. Aynı elin altındaki dört düğüm, o elin dürüst olduğu varsayımından fazla ne üretir?

Cevap, kesinti dayanıklılığıdır. Bir düğüm çökerse diğerleri çalışır. Ama güven açısından, tek işletmecinin dört düğümlü zinciri ile tek işletmecinin imzaladığı, özet zincirli bir günlük aynı varsayıma dayanır: "bu işletmeci yalan söylemiyor". İkincisi ucuzdur, standarttır ve AB ekosisteminin kendi modelidir. ADR-0009 bu gözlemi şöyle bağladı: zincir ancak **en az iki bağımsız onaylayıcı işletmecisi**, yani geçici işletmeciden hukuken ve operasyonel olarak bağımsız en az bir kurum, onaylayıcı işletmeyi yazılı olarak kabul ettiğinde kurulur (K4). O güne kadar güvenin dayanağı imzalı listeler ve çapa günlüğüdür.

TARİHÇE

Değerlendirilen dört yol

ADR-0009 kararı verilirken dört seçenek tartıldı [3]. Zinciri beklemek, pilotu aylarca bir altyapı kurulumuna bağlıyor ve güvence kazancı sağlamıyordu. Tek düğümlü ya da dört düğümlü bir "geliştirme zinciri" ile pilota çıkmak zincir görünümü verip güven vermeyecekti. Güven listesi, sertifika ve iptal listesi olmayan basit bir demo aynı mimari olmadığı için geçişte her şeyin yeniden yazılmasını gerektirecekti. İlk günden ikinci bir kurumla çok taraflı zincir kurmak doğru hedefti, ama ikinci bağımsız işletmeci o gün yoktu; onu bulmanın yolu pilotu göstermekti. Seçilen yol bu sonuncusunun kapısını açık tutan imzalı listelerdi.

Listeler bir ara çözüm gibi görünebilir, ama tasarımın kendisi zincire göre kurulmuştur. Liste biçiminin her alanı bir kontrat kaydına karşılık gelir (SPEC-TRUST-0001 §1) [13]. Listeler ilk günden çok devletli bir ağ varsayılarak düzenlenmiştir: listelerin listesinde Türkiye'nin yanında başka Türk devletleri için de yuvalar bulunur ve bu yuvalar bugün boştur (`RESERVED`).

ÖNEMLİ

Devlet yuvaları hakkında

Listelerde Türkiye dışındaki devletler için ayrılmış yuvalar bir teknik hazırlıktır. Bu devletler için yer ayrılmıştır; anlaşma ya da görüşme bildirimi değildir. Bugün yalnız Türkiye listesi etkindir ve o listeyi de Tamga Network ulusal makam adına, geçici işletmeci olarak yayımlar.

3 Zincir Neyi Ekler, Neyi Eklemez?

3.1 Çift konuşma

Timur'un birinci sorusuna dönelim. Bugün listeleri tek bir işletmeci imzalıyor. Bu işletmeci bir doğrulayıcıya bir liste, başka bir doğrulayıcıya başka bir liste gösterirse ne olur? Literatürde bu, **bölünmüş görüş** (split view) ya da Tamga belgelerindeki adıyla **çift konuşma** olarak bilinir. Örneğin iptal edilmiş bir kurumun belgeleri bir doğrulayıcıda kabul edilirken başka bir doğrulayıcıda reddedilebilir.

Tamga bu zayıflığı saklamaz; ağın kendi belgelerinde "bilinen zayıflama" olarak yazılıdır (ADR-0009 K3) [3]. Herkese açık çapa günlüğü, değişiklik kaydı, üç aylık şeffaflık raporu ve bağımsız denetim çift konuşmayı **caydırır**: farklı listeler gören iki doğrulayıcı özetleri karşılaştırırsa fark ortaya çıkar. Ama bu önlemler çift konuşmayı **imkânsız kılmaz**: işletmeci ile bir belge veren birlikte hareket ederse, kimse karşılaştırma yapmadıkça fark görünmeyebilir. İnternetin sertifika şeffaflığı günlükleri aynı soruna benzer bir cevap verir; tek bir günlük de çatallanabilir ve bunu yakalamak için birden çok bağımsız gözlemcinin aynı görüntüyü paylaşması gerekir [22].

3.2 Defterin ekleyebileceği iki şey

Bir defter bu tabloya iki şey ekleyebilir. Birincisi **birden çok bağımsız tarafın onayladığı değiştirilemez bir geçmiştir**. Bir liste sürümünün özeti deftere ancak birden çok bağımsız işletmeci aynı özeti gönderince "onaylı" olarak yazılırsa, tek bir işletmeci iki farklı liste göstermek için artık diğerlerini de ikna etmek zorunda kalır. Çift konuşma "caydırılan" bir şey olmaktan çıkar, uzlaşmayla engellenen bir şeye dönüşür.

İkincisi **çok taraflı yönetişimdir**. Bugün ağın kurallarını değiştirmek bir işletmecinin ve proje yönetiminin kararıdır. Defterde ise üyelik, ortak belge türleri ve kuralların değiştirilmesi devletlerin üçte iki oyuna bağlanır ve bu kurallar belgede değil kodda yaşar (Bölüm 11).

Bu iki şeyin ortak noktası, ikisinin de **imzacı sayısına** bağlı olmasıdır. Defterin değerini verinin nerede durduğu belirlemez; kaç bağımsız tarafın onu onayladığı belirler. Tamga'nın kurallarında ortak defterin bir depolama tercihi sayılmayıp bir imzacı tercihi olarak tanımlanması bu yüzdendir [1].

3.3 Defterin eklemeyeceği şeyler

Bir defterin eklemeyeceği şeyler de en az eklediği şeyler kadar önemlidir ve tasarımın sınırlarını çizer.

Kişisel veri asla. Deftere kişinin adı, kimlik numarası, belgenin içeriği ya da belgenin özeti yazılmaz. Bu sınır bir tercih değil, kontratların bağlayıcı kuralıdır (SPEC-BC-0001 DP1) [14]. Bir deftere yazılan şey silinemez; kişisel verinin silinme hakkı ile değiştirilemez bir kaydın çatışması dizinin on ikinci yazısında ayrıntılı anlatıldı [2]. Tamga'nın cevabı çatışmayı hiç doğurmamaktır.

Doğrulamada bağımlılık yok. Doğrulama hiçbir adımda deftere bağlı değildir ve olmayacaktır (ADR-0051 CH1) [5]. Defter erişilemese de Aynur'un diploması bugünkü gibi doğrulanır.

Belge başına kayıt yok. Bir belge verildiğinde deftere hiçbir şey yazılmaz. Belgenin verilmesi ve gösterilmesi kurumla kişi, kişiyle doğrulayıcı arasında olur. Defter yalnız kurumları, yetkileri ve listelerin özetlerini bilir (SPEC-BC-0001 §11.1) [14].

Hukuki tanıma yok. Bir defter, bir belgenin başka bir ülkede hukuken geçerli sayılmasını sağlamaz. Bu devletler arasındaki anlaşmaların konusudur.

Tablo 1. Defterin güven ağına ekledikleri ve eklemedikleri.

Konu	İmzalı listeler (bugün)	Ortak defter (açılırsa)
Listenin değiştirilmediğine tanıklık	Tek işletmecinin imzası ve herkese açık günlük	Eşik kadar bağımsız işletmecinin onayı
Çift konuşma	Caydırılır, imkânsız değildir	Onaylı farklı özet uzlaşmayla engellenir; çelişki kayda geçer
Kural değişikliği	İşletmeci ve proje yönetimi	Devletlerin 2/3 oyu, kodda
Doğrulamanın dayanağı	İmzalı liste	Yine imzalı liste; defter isteğe bağlı ek denetim
Kişisel veri	Yok	Yok
Belge başına kayıt	Yok	Yok

4 Defterin Motoru: Besu, QBFT ve İzinli Ağ

4.1 Neden Besu ve QBFT?

Defter bugün kurulmamış olsa da teknolojisi seçilmiştir: Hyperledger Besu ve QBFT uzlaşma yöntemi (ADR-0001) [4]. Besu, Ethereum yazılımını çalıştıran açık kaynaklı bir istemcidir; Apache 2.0 lisanslıdır. Tamga'nın kullanımında Ethereum'a bağlanmaz: kendi başlangıç bloğu (genesis), kendi zincir kimliği ve kendi onaylı onaylayıcı kümesiyle çalışan bağımsız, izinli bir ağ olarak kurulur. Seçimin gerekçeleri izinli yapı, devletler arasında eşit oy gücü, blokların anında kesinleşmesi, egemenlik ve Avrupa'daki emsaldir; Avrupa'nın kamu blokzincir altyapısı EBSI de Besu tabanlıdır. Değerlendirilen başlıca alternatif Cosmos SDK idi; çok zincirli yapının avantajı tek ortak defterde işe yaramayacağı için seçilmedi.

QBFT, Besu'nun yetki kanıtına (proof of authority) dayanan Bizans hatasına dayanıklı uzlaşma yöntemlerinden biridir ve IBFT 2.0'ın devamıdır [18, 19]. "Bizans hatası" adını, bir kısmı hain olabilecek generallerin ortak bir karara varması problemini anlatan klasik makaleden alır [20]: katılımcıların bir kısmı çökebilir, susabilir ya da bilerek yanlış mesaj gönderebilir, yine de dürüst olanlar aynı sonuca varmalıdır.

TEKNİK AYRINTI

Kaç onaylayıcı, kaç hata?

Bizans hatasına dayanıklı yöntemlerde n onaylayıcının en çok f tanesi hatalı ya da kötü niyetli olabilir ve sistem yine doğru çalışır, yeter ki aşağıdaki koşul sağlansın [20, 18]. Bir blok, onaylayıcıların üçte ikisinden fazlasının imzasıyla tek adımda kesinleşir; Tamga'nın kurulum belgesindeki örnekte dört onaylayıcıyla bir hatalı düğüme dayanılır [29]:

$$n \geq 3f + 1$$
$$q = 2f + 1$$
$$n = 4 \implies f = 1, \quad q = 3$$

q bir bloğun kesinleşmesi için gereken imza sayısıdır. Buradaki dayanıklılık bir varsayıma bağlıdır: hatalı olabilecek düğümler birbirinden bağımsızdır. Dört düğümün hepsini aynı taraf işletiyorsa o taraf tek başına f'yi aşar ve formül anlamsızlaşır. Bölüm 2.4'teki "ikinci bağımsız işletmeci" kuralının matematik karşılığı budur.

Tasarımda onaylayıcılar devletlerdir; güvenilir kurumlar blok onaylamayan tam düğümler (full node) işletebilir. QBFT birkaç düzine onaylayıcının üzerinde zorlandığından bu ayrım ölçek sorununu da çözer [4].

4.2 İzinli ağ: kim bağlanır, kim yazar?

Defter izinlidir: yalnız onaylı düğümler ağa bağlanır, yalnız onaylı hesaplar işlem gönderir. Tamga'nın ilk belgeleri bunu Besu'nun zincir üstü izin kontratıyla yapmayı öngörüyordu. Besu bu özelliği 24.12 sürümünde kullanımdan kaldırdı ve 25.6.0 sürümünde tümden çıkardı [21]. ADR-0052 kalan yolu seçti: her düğümün kendi klasöründe duran yerel bir izin dosyası (`permissions_config.toml`) [6].

Bu dosyanın bir zayıf yanı vardır: Besu, yerel hesap kurallarını başka bir düğümün ürettiği bloklara uygulamaz; yani onaylayıcıların listeleri farklıysa biri, diğerlerinin reddedeceği bir hesaptan gelen işlemi blokuna koyabilir. ADR-0052 bunu bir kuralla kapatır: izin dosyası **elle yazılmaz**, defterin kendi kayıtlarından üretilir. Hangi hesapların işlem gönderebileceği (aktif devletlerin onaylayıcı adresleri ve delege anahtarları, çalışır durumdaki kurumların delege anahtarları, liste çapası yazıcıları) zincirdeki kontratlarda zaten yazılıdır. Her işletmeci dosyayı kendisi üretir, özetini diğerleriyle karşılaştırır; fark bir işletme olayıdır ve giderilene kadar raporlanır.

Bir ayrım daha var. İzin listesi ikinci savunma hattıdır; yazma yetkisinin asıl denetimi kontratlardadır. İzin listesindeki bir hata, bir hesaba kontratın vermediği bir yetkiyi vermez (LP4) [6]. Düğümlerin uzaktan çağrı arayüzü (JSON-RPC) internete açılmaz; dış erişim yalnız karşılıklı TLS doğrulayan bir ters vekil arkasından ve tanımlı istemcilere verilir. Yönetim ve izin ad alanları yalnız düğümün kendisinden kullanılabilir (LP5).

Ücretsiz gaz ve kotalar. İzinli ağda işlem ücreti (gas) sıfırdır. Bu, yazmanın serbest olduğu anlamına gelmez. Kontratların bir kuralı bunu açıkça söyler: deftere yazılan her bayt her onaylayıcının diskinde kalıcıdır; ücretsiz olması maliyetsiz olduğu anlamına gelmez (GA1) [14]. Bu yüzden kontratlar kendi içinde günlük kotalar tutar.

Tablo 2. Kontrat içi günlük kotalar (SPEC-BC-0001 §12.2; ADR-0054 NS3).

Kota	Değer	Kapsam
Devletin ulusal yazımı	Günde 10.000	Devlet başına, her kayıt kontratında ayrı sayılır
İptal listesi yayını	Günde 48	Liste başına (saatlik yayın ve yeniden deneme payı)
Yeni iptal listesi kaydı	Günde 8	Kurum başına; yayın kotası yeni liste açılarak aşılamaz

Sayaçlar her gün UTC gün başında sıfırlanır. Bu kotaların geçmişin deftere aktarıldığı tek seferlik göç penceresinde neden askıya alındığını Bölüm 10'da göreceğiz.

5 Kontratların Haritası

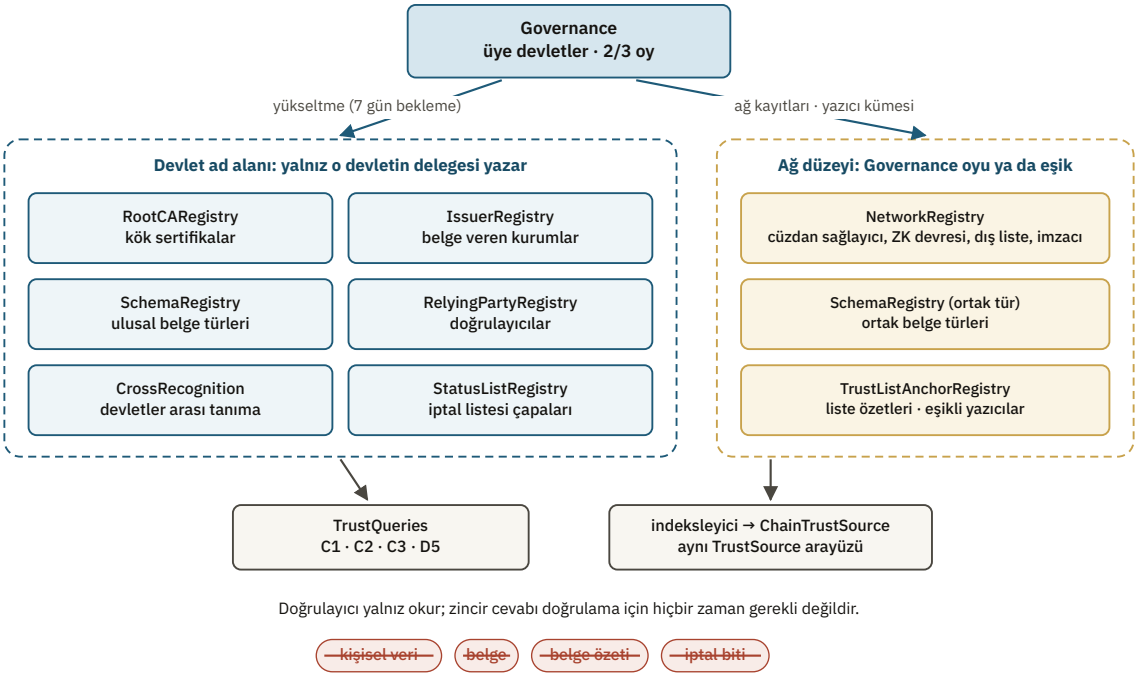
5.1 Dokuz kontrat ve bir okuma kapısı

Defterin kuralları Solidity dilinde yazılmış akıllı sözleşmelerdedir (bu yazıda kısaca **kontrat**). Kontratlar Hardhat ile derlenip test edilir; derleyici solc 0.8.24, hedef EVM sürümü Shanghai, kütüphane OpenZeppelin 5.6'dır (SPEC-BC-0001 §15) [14]. Tablo 3 kontratları ve her birinin imzalı listelerdeki karşılığını gösteriyor.

Tablo 3. Kontratlar ve listelerdeki karşılıkları.

Kontrat	Ne tutar	Kim yazar	Listedeki karşılık
Governance	Üye devletler, onaylayıcı adresleri, delege anahtarları, öneriler ve oylar	Devletlerin onaylayıcıları (öneri ve oy)	Listelerin listesindeki ülke yuvaları
RootCARegistry	Ulusal kök sertifikalar ve durum geçmişleri	O devletin delegesi	`root_cas[]`
IssuerRegistry	Belge veren kurumlar, belge türü yetkileri, durum geçmişleri	O devletin delegesi	`issuers[]`
SchemaRegistry	Belge türleri ve içerik özetleri	Ulusal tür: o devlet; ortak tür: Governance oyu	`schemas[]`, `national_schemas[]`
RelyingPartyRegistry	Doğrulayıcılar, kullanım kapsamları, kapı grupları	O devletin delegesi	`relying_parties[]`
CrossRecognition	Hangi devletin hangi devletin kurumlarını tanıdığı	Her devlet kendi tanımasını	Ülke yuvasının tanıma alanı
StatusListRegistry	İptal listelerinin çapaları (adres, özet, sürüm)	Kurumun delegesi	Çapa günlüğündeki iptal listesi satırları
TrustListAnchorRegistry	Liste sürümlerinin ve çapa günlüğü başının özetleri	Eşikli yazıcılar	Liste dosyalarının kendisi
NetworkRegistry	Ağ düzeyi kayıtlar (cüzdan sağlayıcılar, ZK devreleri, dış listeler, imzacılar, işletmeci)	Governance oyu	Listelerin listesinin ağ düzeyi alanları
TrustQueries	Yazmaz; doğrulama sorularını tek yerden cevaplar	Kimse	`TrustSource` arayüzünün zincir karşılığı

Kayıt kontratları yükseltilebilir yapıdadır ve UUPS deseniyle (ERC-1822, ERC-1967) kurulur; yükseltme yetkisi yalnız `Governance` kontratındadır [14, 24]. Depolama düzeni değişiklikleri yalnız sona ekleme biçiminde yapılır ve bu, her derlemede bir karşılaştırma testiyle denetlenir.



Şekil 2. Kontratların haritası: devlet ad alanları, ağ düzeyi kayıtlar ve okuma kapısı

5.2 Egemenlik kodda

Ağın yönetişim kararlarından biri, ulusal kayıtların yalnız ilgili devletin yetkisinde olmasıdır (ADR-0002) [23]. Defterde bu kural her yazma fonksiyonunun başındaki bir denetim olarak durur. Bir devletin delege anahtarı başka bir devletin ad alanındaki hiçbir kaydı değiştiremez; denerse işlem geri çevrilir.

SOLIDITY Örnek: egemenlik denetimi (SPEC-BC-0001 §0)

```
/// stateCode: ISO 3166-1 alpha-2 (bytes2)
modifier onlyOwnerState(bytes2 stateCode) {
    if (!_isDelegateOf(msg.sender, stateCode)) revert NotOwnerState(stateCode, msg.sender);
    _;
}
```

Bu kuralın tek istisnası devletlerin ortak oyudur: ortak belge türleri, protokol yükseltmesi ve türü belli kayıt yönetimi işlemleri yalnız `Governance` üzerinden, üçte iki oyla yapılır (N1) [14]. ADR-0054 kuralın kenarlarını da kapattı: bir kurum yalnız kendi devletinin kök sertifikasına bağlanabilir ve yalnız kendi devletinden bir halef gösterebilir; ulusal belge türlerinin adı `urn:tamga:<cc>:…` biçiminde kaydı yapan devletin koduyla başlamak zorundadır, ortak türler bu biçimde olamaz [8]. Böylece bir devlet, oylanmakta olan ortak bir türün adını önceden kendi adına kapamaz.

5.3 Doğrulamanın beş sorusu

Bir doğrulayıcının zincire sorduğu sorular, bugün listeye sorduğu sorularla aynıdır. Kontrat şartnamesi bunları beş okuma olarak tanımlar (SPEC-BC-0001 §11.2) [14]:

SOLIDITY Örnek: doğrulama okuma seti (SPEC-BC-0001 §11.2)

```
// C1: iat anında bu kurumun belgesi kabul edilebilir miydi
issuerRegistry.isCredentialAcceptable(issuerId, credentialIat)
// C2: iat anında kurum bu belge türüne yetkili miydi
issuerRegistry.isCredentialSchemaAcceptable(issuerId, schemaId, credentialIat)
// Şema bütünlüğü: belge türü tanımının özeti kayıtlı mı
schemaRegistry.matchesContentHash(schemaId, typeMetadataHash)
// C3: doğrulayıcının devleti bu kurumu tanıyor mu
crossRecognition.isRecognizedBy(verifierState, issuerId)
// D5: iptal listesinin çapası (özet ve sürüm doğrulayıcı tarafında karşılaştırılır)
statusListRegistry.getListAnchor(listId)
```

Bu yazının geri kalanında **C1, C2, C3 ve D5** adları bu sorular için kullanılıyor. Dikkat edilirse hiçbir soru "bu belge iptal edildi mi" diye sormaz. İptal bitleri deftere hiç yazılmaz; iptal durumu, kurumun imzaladığı ve her doğrulayıcının aynı biçimde indirdiği iptal listesinden okunur [25]. Defter yalnız o listenin özetini ve sürümünü bilir. Böylece hangi belgenin ne zaman iptal edildiği defterde iz bırakmaz.

TEKNİK AYRINTI

Doğrulayıcı zincire doğrudan sormaz

Beş okuma, her doğrulamada bir defter düğümüne istek göndermek anlamına gelseydi hem yavaş hem de izlenebilir olurdu. Bu yüzden tasarımda zincir doğrudan sorgulanmaz. Bir **indeksleyici** kontratların yayınladığı olayları (event) izler, değişen kayıtları aynı sabitlenmiş bloktaki okumalarla yerel bir dizine yazar; `ChainTrustSource` adlı okuma kaynağı da `TrustSource` arayüzünü bu dizinden cevaplar [14, 11]. Dizin bayatsa ya da zincire erişilemiyorsa cevap "bilinmiyor"dur. Cüzdan ve tarayıcı çekirdeği zincir istemcisi taşımaz; ağın paketleri zincir bağımlılığı almaz [15]. **Durum:** indeksleyici ve `ChainTrustSource` yerel geliştirme ağında çalışıyor.

6 Projeksiyon İlkesi ve Liste Çapası

6.1 Kaynak liste, türev defter

Defter açılırsa listelerin yerine mi geçecek? ADR-0051 bu soruya en dar ve en güvenli cevabı verir: **zincir, imzalı listelerin ve çapa günlüğünün projeksiyonudur, kaynağı değildir** [5]. "Projeksiyon" burada bir nesnenin bir yüzeye düşen gölgesi gibi düşünülebilir: gölge nesnenin biçimini taşır, ama nesneyi değiştirmek için gölgeye dokunmak işe yaramaz. Listeler ne diyorsa doğru olan odur; defter yalnız "bu liste sürümü şu anda, şu özetle yayımlandı" diye tanıklık eder.

Kararın bağlayıcı kuralları beş maddedir (ADR-0051 CH1–CH5) [5]:

1. Zincir imzalı listelerin ve çapa günlüğünün projeksiyonudur; çelişkide imzalı liste geçerlidir; zincirin cevabı doğrulama için gerekli değildir ve "bilinmiyor" cevabı sonucu kötüleştirmez.
2. Liste çapası kaydına yalnız özetler, sürüm ya da sıra numaraları ve zamanlar yazılır; liste içeriği, belge, belge özeti ya da kişisel veri yazılmaz.
3. Bir çapa ancak onay eşiği kadar farklı yetkili yazıcı birebir aynı veriyi gönderince onaylıdır; üretimde eşik en az ikidir.
4. Sürüm numaraları yalnız artar; onaylı kayıt silinmez ve değiştirilmez.
5. Özellik bayrağı (`TAMGA_CHAIN_ENABLED`) kapalıyken zincire bağlanılmaz; üretimde bayrak ikinci bağımsız işletmeci ve en az iki eşik olmadan açılmaz.

6.2 Zincire yazılan iki kayıt

Liste çapası kaydı (`TrustListAnchorRegistry`) yalnız iki tür kayıt tutar. Birincisi her liste sürümünün özetidir: hangi liste, kaçıncı sürüm, içeriğin SHA-256 özeti, önceki sürümün özeti ve yayın zamanı. İkincisi çapa günlüğünün son satırının özetidir. Günlüğün satırları birbirine özet zinciriyle bağlı olduğundan son satırın özeti bütün günlüğü bağlar; her satırı ayrı ayrı çapalamaya gerek kalmaz.

SOLIDITY Örnek: liste çapası kaydının yazma fonksiyonları (ITrustListAnchorRegistry)

```
struct ListAnchor {
    uint64  version;
    bytes32 contentHash;   // sha256(<liste>.jws), sonraki sürümün previous_version_hash değeri
    bytes32 previousHash;  // önceki sürümün özeti (ilk sürümde 0)
    uint64  issuedAt;      // listenin issued_at'i (epoch saniye)
    uint8   confirmations;
    bool    confirmed;
    uint64  confirmedAt;
}
function anchorList(bytes32 listKey, uint64 version, bytes32 contentHash,
                    bytes32 previousHash, uint64 issuedAt) external;
function anchorLogHead(bytes32 logKey, uint64 seq, bytes32 lineHash, uint64 ts) external;
```

Kayıtta metin alanı yoktur; bütün alanlar sabit boyutlu özetler ve tam sayılardır. Listenin hangi liste olduğu da bir metinle değil, bir özetle belirtilir:

$$\begin{aligned} \text{listKey} &= \text{keccak256}(\text{tamga:trust-list:} \parallel \text{ortam} \parallel : \parallel \text{dosya}) \\ \text{contentHash} &= \text{SHA-256}(\text{dosya.jws}) \\ \text{logKey} &= \text{keccak256}(\text{tamga:anchors:} \parallel \text{ortam}) \end{aligned}$$

Burada $\parallel$ metinlerin art arda eklenmesidir; “ortam” gerçek ağ için `production`'dır. `keccak256`, Ethereum'un kullandığı ve SHA-3 standardından önceki dolgu kuralını taşıyan Keccak özet işlevidir [26].

6.3 Eşikli yazıcılar ve rakip adaylar

Bir özeti deftere tek bir yazıcının (anchorer) göndermesi yetmez. Liste yayıncısı her liste yayınından ve her saatlik çapa satırından sonra kendi çıktısını önce doğrular (imza, özet zinciri, tazelik; sağlıksız set çapalanmaz), sonra yalnız özetleri gönderir. Bu gönderim bir **oydur**. Bir sürüm, eşik kadar farklı ve hâlâ yetkili yazıcı aynı özeti destekleyince onaylanır.

Yazıcılar farklı özetler gönderirse ne olur? İlk tasarımda sürümü ilk açan yazıcı verinin ne olacağını belirliyor, farklı veri gönderen reddediliyordu. Bir güvenlik incelemesi bunun iki zayıflığını gösterdi: tek bir yazıcı bir sürümü yanlış veriyle açıp dürüst yazıcıları kalıcı olarak engelleyebiliyordu ve ret zincirde iz bırakmadığı için bu kilit görünmüyordu. ADR-0053 kuralı değiştirdi: aynı sürüm için her farklı veri ayrı bir **aday** olarak kaydedilir, ikinci ve sonraki her farklı aday herkesin görebileceği bir çelişki olayı (`AnchorConflict`) yayar ve eşiği ilk dolduran aday kazanır [7].

Aynı akış adım adım şöyle işler (D bir adayı destekleyen yazıcılar, k onay eşiği; kesin kural aşağıdaki kutuda):

1	`Liste yayıncısı`	**Yayın.** Yeni sürümü imzalar; önceki sürümün özetini içine yazar.	$\text{prev}_n = \text{SHA-256}(L_{n-1})$
2	`İşletmeci A`	**Öz denetim.** Kendi indirdiği listeyi doğrular; sağlıksızsa göndermez.	$\text{Verify}(L_n) = \text{geçerli}$
3	`İşletmeci A → Defter`	**Oy.** Yalnız özeti gönderir; yeni bir aday açılır.	$h_n = \text{SHA-256}(L_n), \quad \lvert D\rvert = 1$
4	`İşletmeci B → Defter`	**İkinci oy.** Bağımsız olarak aynı özeti hesaplar ve gönderir.	$\lvert D(n, h_n)\rvert = 2 \geq k$
5	`Defter`	**Onay.** Eşik dolunca sürüm kapanır; kayıt artık değişmez.	$\text{onaylı}(n, h_n)$
6	`Doğrulayıcı`	**İsteğe bağlı denetim.** Elindeki listenin özetini onaylı çapayla karşılaştırır.	$h \stackrel{?}{=} h_n$

TEKNİK AYRINTI

Eşikli onayın kuralı

W yetkili yazıcılar kümesi, k onay eşiği, $D(v,a)$ de v sürümü için a adayını destekleyen ve hâlâ yetkili olan yazıcılar olsun (ADR-0051 CH3; ADR-0053 AC1–AC4) [5, 7]:

$$\text{onaylı}(v,a) \iff |D(v,a)| \geq k$$
$$2 \leq k \leq |W| \quad \text{(üretimde)}$$
$$D(v,a) \cap D(v,b) = \varnothing \quad \text{her } a \neq b \text{ için}$$

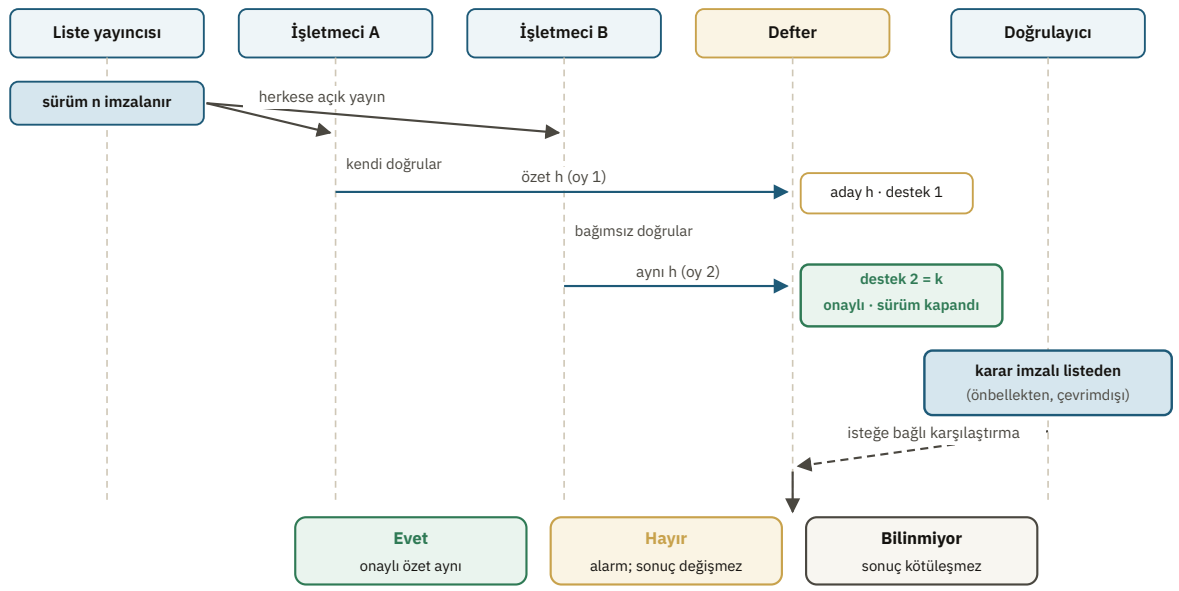
Üçüncü satır bir yazıcının aynı sürüm için iki adayı destekleyemeyeceğini söyler. Aksi hâlde tek bir işletmeci iki farklı özete destek verip çift konuşmayı kendi içinde gizleyebilirdi. Kümeden çıkarılan yazıcının desteği sayılmaz; böylece anahtarını değiştiren bir işletmeci aynı sürüme eski ve yeni anahtarıyla iki kez oy veremez. Eşik de yazıcı sayısını aşacak biçimde ayarlanamaz; aksi hâlde hiçbir çapa onaylanamazdı.

6.4 Doğrulayıcının üç cevabı

Doğrulayıcı tarafındaki denetim (`trust:chain-check`) elindeki imzalı listenin özetini hesaplar ve defterdeki onaylı çapayla karşılaştırır. Üç cevaptan biri çıkar [5]:

- **Evet (`YES`):** onaylı çapa elindeki listenin özetiyle aynıdır.
- **Hayır (`NO`):** aynı sürüm için onaylı ama **farklı** bir özet vardır. Bu, iki doğrulayıcıya iki farklı liste gösterildiğinin işaretidir. Doğrulama sonucunu değiştirmez, ama işletmeciye alarm üretir ve şeffaflık raporuna girer.
- **Bilinmiyor (`UNKNOWN`):** özellik kapalıdır, zincire erişilemiyordur, çapa yoktur ya da eşik henüz dolmamıştır. Sonuç kötüleşmez.

"Hayır" cevabının doğrulama sonucunu değiştirmemesi ilk bakışta tuhaf görünebilir. Ama projeksiyon ilkesinin doğal sonucudur: çelişkide imzalı liste geçerlidir. Defter bir hakem değil, bir tanıktır; tanığın gördüğü çelişki bir alarm üretir, kararı tek başına değiştirmez. Defter kaydı da düzeltilmez; yanına yeni bir sürüm yazılır.



Şekil 3. Projeksiyon: kaynak liste, iki bağımsız oy, isteğe bağlı denetim

Durum: Liste çapası kaydı, `trust:chain-submit` ve `trust:chain-check` komutları yerel geliştirme ağında **çalışıyor**. Bu ağ tek onaylayıcılı bir Besu QBFT ağıdır ve yalnız geliştirme makinesinde, herkesçe bilinen test hesaplarıyla çalışır; üzerindeki duman testinde listeler çapalandı ve denetim "evet" cevabı verdi [29]. Üretimde bayrak kapalıdır; üretim kullanımı **planlanan** durumdadır.

7 "Her Şey Zincirde": Kapsam ve Kişisel Veri Sınırı

7.1 Liste özetinden kayıt düzeyine

Projeksiyonun ilk biçimi yalnız liste özetlerini yazıyordu. Bu, çift konuşmayı yakalamak için yeterlidir ama defterden "bu kurum şu tarihte bu belge türüne yetkili miydi" diye sormaya yetmez. Kayıt kontratları (Tablo 3) bu soruları cevaplamak için vardı, ama listelerin bazı alanlarının defterde hiç karşılığı yoktu: cüzdan sağlayıcılar, kapı grupları, dış listeler, sıfır bilgi ispatı (ZK) devreleri, kayıt sertifikalarının durum listesi, liste işletmecisi ve kurumların kayıt bilgileri.

Proje yönetiminin 4 Kasım 2026'daki yönü açıktı: kapsam "her şey zincirde" olacak. ADR-0074 bunu bir ilkeye bağladı: imzalı listelerde bir güven kararını etkileyen **her kayıt**, defterde bir kontrat kaydına ya da bir çapaya eşlenir (PJ1) [11]. Liste yine kaynaktır ve çelişkide liste geçerlidir.

7.2 Sınır: özet yazılır, metin yazılmaz

"Her şey zincirde" ifadesi kolayca yanlış okunabilir. Bir kurumun kayıt bilgilerinde vergi numarası, posta adresi, iletişim adresi, hizmet açıklaması gibi alanlar vardır. Bunlar kişisel veri olmasa da serbest metindir ve deftere bir kez yazılan metin silinemez. ADR-0074'ün ikinci kuralı bu yüzden kesindir: deftere yalnız tanımlayıcı özetleri, parmak izleri, kapalı küme değerleri, sayılar, tarihler, devlet kodları ve adresler yazılır. Serbest metin ve kayıt ayrıntısı yalnız **ayrıntı özeti** olarak girer (PJ2) [11].

TEKNİK AYRINTI

Ayrıntı özeti nasıl hesaplanır?

Listedeki ilgili alan bloğu B önce kanonik JSON biçimine (RFC 8785, JCS) çevrilir, sonra SHA-256 ile özetlenir [27, 11]:

$$\mathrm{detailsHash} = \mathrm{SHA\text{-}256}(\mathrm{JCS}(B))$$

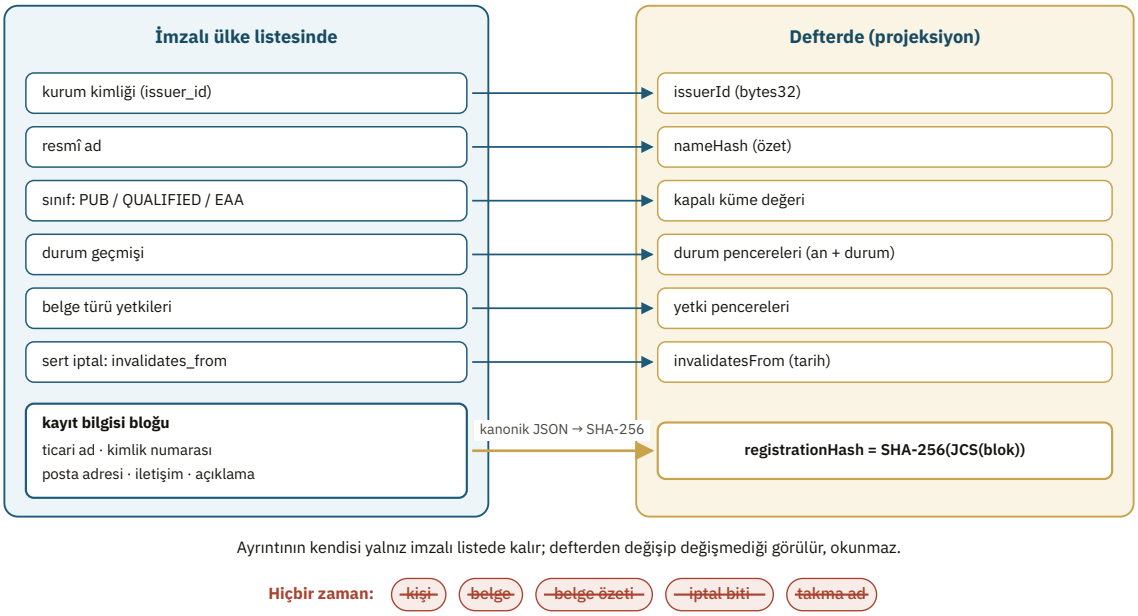
Kanonik biçim aynı JSON nesnesinin her yazılımda aynı bayt dizisine dönüşmesini sağlar: anahtarlar belirli bir sırayla dizilir, boşluklar atılır, sayılar tek bir biçimde yazılır. Bu olmadan iki dürüst taraf aynı kaydı farklı baytlarla özetleyip farklı özetler bulabilirdi. Özetin işi şudur: defterden bakan biri, listedeki ayrıntının değişip değişmediğini görebilir ama ayrıntının kendisini defterden okuyamaz. Ayrıntı yalnız imzalı listede durur.

Aynı yaklaşım kontratların arayüzünde de bir kurala dönüşür: yeni alanlar yalnız `bytes32`, tam sayı, `bool`, `bytes2` ya da `address` olabilir; yeni metin (`string`) alanı ve serbest `bytes` alanı eklenmez. Bu bir test olarak yaşar. Her derlemede bütün kontratların arayüzü taranır; bir alanın adı kişisel veri ya da belge anlamı taşıyorsa (kişi, doğum, kimlik numarası, belge, beyan, sahip gibi) ya da izin verilen birkaç kurumsal adres alanı dışında bir metin alanı varsa test kırılır [14].

Tablo 4. Bazı liste alanları ve defterdeki karşılıkları (ADR-0074 K2–K4).

Liste alanı	Defterde	Defterde olmayan
Kurum kayıt bilgisi (ticari ad, kimlik numaraları, posta adresi, iletişim, hizmet açıklaması, denetim kurumu)	Kayıt verisi özeti; kamu kurumu olup olmadığı (`bool`); yetki bit maskesi	Metinlerin kendisi

Liste alanı	Defterde	Defterde olmayan
Kurum sınıfı	Kapalı küme: `PUB`, `QUALIFIED`, `EAA`	
Doğrulayıcının kullanım kapsamları	Kapsam ve hizmet özetleri, belge türü, biçim, öğe yolu özetleri, geçerlilik aralığı	Amaç metni, gizlilik politikası adresi (ayrıntı özeti olarak)
Kapı grupları	Grup kimliği özeti, kurum bağı, politika özetleri, kapı sertifikası parmak izleri, çevrimdışı izni	Grup etiketi (özet olarak)
Cüzdan sağlayıcılar	Sağlayıcı kimliği özeti, ad özeti, durum geçmişi, imza anahtarı parmak izleri	
ZK devreleri	Devre kimliği, sistem, sürüm, öznitelik sayısı, devre dosyasının SHA-256'sı	Açıklama notu
Dış listeler	Liste kimliği, bölge kodu, biçim, sabitlenmiş imzacı parmak izleri, kapsam	Liste adresi (ayrıntı özeti olarak)
Liste işletmecisi	Liste anahtarı, ad özeti, durum (geçici, atanmış)	Adın kendisi
Her durumda		Kişi, belge, belge özeti, iptal biti, takma ad, cihaz ya da cüzdan kimliği



Şekil 4. Bir kurum kaydı listede ve defterde: ayrıntı imzalı listede kalır, deftere özeti gider

7.3 Ağ düzeyi kayıtlar ve yeni günlük satırları

Cüzdan sağlayıcılar, ZK devreleri, dış listeler ve liste imzacıları hiçbir devletin kaydı değildir; ağın ortak kaydıdır. Bu yüzden mevcut ulusal kontratlara dağıtılmadılar ve yeni bir ağ düzeyi kontratta (`NetworkRegistry`) toplandılar. Bu kontrata yalnız `Governance`'ın türü belli önerileriyle, üçte iki oyla yazılır; tek istisna Bölüm 10'daki göç penceresinde yetkilendirilen göç yürütücüsüdür (PJ3) [11].

Bir kaydı oylamak ile onu yazmak arasında bir zaman farkı vardır ve bu fark bir saldırı yüzeyi açabilir: bir öneri oylanırken kaydın kendisi değişirse, oylanan şey ile yazılan şey ayrışabilir. Tasarım bunu iki adımla çözer. Önce kaydın içeriği sabitlenir (`stageRecord`) ve bir kimlik alır; bu kimlik içeriğin ve kaydın o anki yazım sayacının özetidir. Oylanan şey bu kimliktir. Yürütme anında sayaç değişmişse öneri bayattır ve yazılmaz (`StaleStage`) [14].

Listelerdeki iki şey de bugüne kadar çapa günlüğünde yoktu: kayıt sertifikalarının durum listesi ve listelerin ETSI biçimindeki görünümleri. ADR-0074 bunlar için iki yeni günlük satırı türü tanımladı

(`registration_status`, `list_view`) [11]. Bu satırlar doğrulama kararına girmez. Doğrulama kodu bilinmeyen bir satır türü görünce durduğu için, yayıncı bu satırları yalnız zincir projeksiyonu açıkken yazar; önce yeni türleri tanıyan paket sürümü yayımlanır, sonra satırlar gelir [13].

Test ağı deftere girmez. Tamga'nın bir de test ağı (sandbox) vardır. Bu ağın listeleri ve çapa günlüğü her gece sıfırlanır. Defterde ise sürüm numaraları yalnız artar ve hiçbir kayıt silinmez. İkisi bir araya gelemez; test ağının listeleri gerçek defterin güven kaydına karışmamalıdır. Kural kesindir: test ağının listeleri üretim defterine yazılmaz ve yeniden oynatılmaz (PJ4); yazma komutları test ağı setini üretim defterinde reddeder. Test ağı yalnız yerel geliştirme zincirinde denenebilir [11].

Durum: `NetworkRegistry`, kayıt verisi özetleri, kapı grupları ve yeni çapa satırları yerel geliştirme ağında **çalışıyor** üretimde **planlanan**.

8 Belgenin Verildiği Andaki Durum

8.1 Kapanan kurumun eski diploması

Aynur'un diplomasını veren üniversitenin bir gün kapandığını ya da ağdan çekildiğini düşünelim. Aynur'un diploması geçersiz mi olur? Kâğıt dünyasında cevap açıktır: kapanmış bir okulun yıllar önce verdiği diploma, verildiği gün okul yetkili olduğu için geçerlidir. Dijital dünyada aynı cevabı vermek için doğrulayıcının belgenin **verildiği andaki** duruma bakması gerekir; kurumun **bugünkü** durumu bu soruda belirleyici olamaz. Belgenin içindeki verilme anı (`iat`, issued at) bu yüzden doğrulamanın merkezindedir.

İmzalı listeler bunu baştan beri böyle yapar: her kurum kaydı bir durum geçmişi (`status_history`) taşır ve doğrulama bu geçmişe belgenin verilme anıyla bakar [13]. Kontratların ilk hâli ise bazı sorularda bugünkü duruma bakıyordu: askıdaki bir kurumun bütün belgelerini reddediyor, emekliye ayrılmış bir kökün sonraki belgelerini kabul ediyordu. Bir de sınırlarda bir saniyelik fark vardı: liste "iptal anında verilen belge geçersiz" derken kontrat aynı anı geçerli sayıyordu. ADR-0073 bu farkları kapattı ve kuralı listeyle eşitledi (CE5) [10].

> TEKNİK AYRINTI
>
> **`iat` anındaki durum ve yarı açık aralıklar**
>
> Bir kaydın durum geçmişi, yalnız eklenebilen (t_j, s_j) değişikliklerinden oluşur: t_j anından itibaren durum s_j'dir. Her durum bir **yarı açık aralıkta** geçerlidir; başlangıç dahil, bitiş hariç [10, 14]:
>
> $$s(t) = s_j \quad \text{eğer } t \in [t_j, t_{j+1})$$
>
> $$\begin{aligned} \mathrm{C1}(i,t) \iff & s_i(t) = \mathrm{ACTIVE} \wedge t \in [v_{\text{başlangıç}}, v_{\text{bitiş}}] \wedge \text{kök}_i(t) = \mathrm{ACTIVE} \\ & \wedge (\mathrm{invalidatesFrom}_i = 0 \vee t < \mathrm{invalidatesFrom}_i) \end{aligned}$$
>
> $\mathrm{C1}(i,t)$, i kurumunun t anında verdiği bir belgenin kabul edilip edilmeyeceğidir. Yarı açık aralık sayesinde bir durumun bittiği an ile sonrakinin başladığı an çakışmaz; askıya alma anında verilmiş bir belge askı süresinde sayılır. Son koşul Bölüm 8.3'teki sert iptaldir.

Bu kuralın sonuçları sade cümlelerle şöyle: askıdan önce verilmiş belge geçerlidir, askı süresinde verilen belge geçersizdir; iptal anında ya da sonrasında verilen belge geçersizdir; emekliye ayrılmış bir kökün emeklilikten sonraki belgeleri geçersizdir; kullanımdan kaldırılan bir belge türüyle kaldırılmadan önce verilmiş belge geçerlidir. Kök sertifikanın iptali ise farklıdır: iptal edilmiş bir kökün altındaki bütün belgeler, ne zaman verilmiş olurlarsa olsunlar düşer [14]. Proje yönetimi bu kuralların ilkini ayrıca onayladı: askıdaki kurumun askıdan önceki belgeleri geçerlidir; zincir listeye uyar [10].

ÖRNEK

Aynur'un diploması ve bir askı

Aynur'un üniversitesi bir yıl Mart ayında Aynur'a diplomasını verdi. Haziran'da kurumun anahtar yönetiminde bir sorun fark edildi ve kurum üç hafta askıya alındı; Temmuz başında yeniden etkin oldu. Askı süresinde kurumun verdiği bir belge (diyelim ki 20 Haziran tarihli bir transkript) doğrulayıcıda reddedilir. Aynur'un Mart tarihli diploması ise askıdan önce verildiği için kabul edilir; Temmuz'dan sonra verilen belgeler de kabul edilir. Doğrulayıcı bu üç kararı, ne üniversiteye ne de işletmeciye sormadan, listedeki durum geçmişine bakarak verir. Defter açıldığında aynı üç karar defterden de aynı çıkmalıdır.

8.2 Durum adlarının eşlenmesi

Listelerde kullanılan durum adları ile kontratların durumları birebir aynı değildir. Listede sekiz değer vardır (`ACTIVE`, `SUSPENDED`, `REVOKED`, `RETIRED`, `ROLLING_OVER`, `RESERVED`, `PROVISIONAL`, `DEPRECATED`); kontratlarda kayıt türüne göre dört ya da beş değer. ADR-0073 bir eşleme tablosu tanımladı [10]. Örneğin listede `RETIRED` olan bir kurum, kontratta bir "yumuşak iptal" olarak yazılır: önceki belgeleri geçerli kalır, yenileri kabul edilmez. Kök sertifikanın `ROLLING_OVER` durumu kontratta `ACTIVE` olarak kalır ve yanına bir "halef duyurusu" olayı eklenir. `RESERVED` ve `PROVISIONAL` ise yalnız boş yuvaları anlatır; bunlar için kurum ya da doğrulayıcı kaydı yazılmaz. Kontratların durum kümelerine yeni değer eklenmedi; ayrım gereken yerde olay kullanıldı.

8.3 Sert iptal: "şu andan sonrasını sayma"

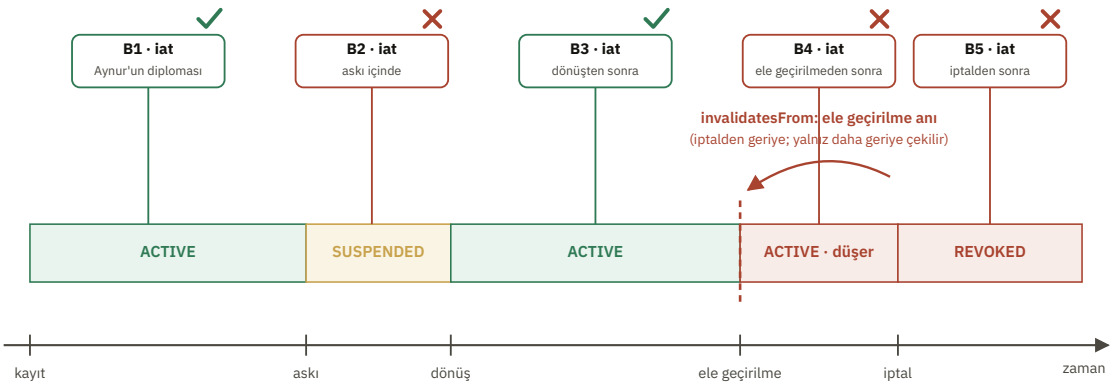
İki tür iptal vardır. **Yumuşak iptal** bir kurumun bundan sonra belge vermemesi demektir; önceki belgeleri geçerli kalır. Bir kurum emekliye ayrıldığında ya da kendi isteğiyle ağdan çekildiğinde doğru olan budur. Ama bir kurumun imza anahtarı ele geçirilmişse durum farklıdır. Saldırgan anahtarı ele geçirdiği andan itibaren kurum adına belge üretmiş olabilir ve bu durum çoğu zaman haftalar sonra anlaşılır. O ara dönemde "verilmiş" görünen belgelerin de düşmesi gerekir.

İmzalı listeler bunu baştan beri bir alanla yapar: iptal kaydına bir **geçersizlik başlangıcı** (`invalidates_from`) yazılır ve o andan sonra verilmiş bütün belgeler düşer [13]. Kontratlarda ise yalnız yumuşak iptal vardı; ele geçirilmiş bir kurumun ara dönemde "verdiği" belgeler defterde kabul edilirdi. ADR-0073 kontrata aynı alanı ekledi (`invalidatesFrom`) [10].

Bu alan, göç penceresi dışında geçmiş bir tarihin yazılabildiği tek alandır ve bu yüzden sıkı sınırlarla çevrilidir (CE2): tarih kurumun geçerlilik başlangıcı ile iptal anı arasında olmalıdır; bir kez yazılan tarih silinmez; yalnız daha geriye çekilebilir. Yani iptalin kapsamı genişleyebilir, hiçbir zaman daralamaz. Bir saldırgan bir delege anahtarını ele geçirse bile bu alanla eski bir iptali geri alamaz ya da kapsamını küçültemez.

$$\mathrm{validFrom}_i \le \mathrm{invalidatesFrom}_i \le \mathrm{revokedAt}_i$$

$$\mathrm{invalidatesFrom}_i^{\text{yeni}} \le \mathrm{invalidatesFrom}_i^{\text{eski}} \quad \text{(yalnız geriye)}$$



Aralıklar yarı açıktır: [başlangıç, bitiş). Karar, kurumun bugünkü durumuna değil belgenin iat anına göre verilir.

B1 ve B3 kabul; B2 askı süresinde, B4 sert iptal kapsamında, B5 iptalden sonra verildiği için ret.

Şekil 5. Belgenin verildiği andaki durum: askı, yeniden etkinleşme ve sert iptal

Durum: `iat` anındaki durum, yarı açık aralıklar, durum eşlemesi ve sert iptal imzalı listelerde ve doğrulama paketinde **çalışıyor** kontratlardaki karşılıkları yerel geliştirme ağında **çalışıyor**.

9 Doğrulayıcı Kimliği: Alan Adı, Devir ve Yeniden Kayıt

9.1 Kimlik sertifikadan alan adına

Bir doğrulayıcının, yani Aynur'dan belge isteyen işverenin, ağda bir kaydı vardır. Bu kayıt hangi belgeleri, hangi alanlarla ve hangi amaçla isteyebileceğini söyler. Cüzdan, Aynur'a "bu işveren bu alanı isteyebilir mi" sorusunun cevabını bu kayıttan verir.

Kaydın kimliği önceden doğrulayıcının erişim sertifikasından türetiliyordu. Bu da bir soruna yol açıyordu: sertifikalar düzenli olarak yenilenir ve her yenilemede kimlik değişir, yani defter açısından yeni bir doğrulayıcı doğar. Takma adlar, açıklama politikaları ve geçmiş kayıtlar ise zaten doğrulayıcının alan adına bağlıydı. ADR-0073 kimliği alan adına bağladı (CE4) [10]:

$$\mathrm{rpId} = \mathrm{keccak256}(\mathrm{utf8}(\mathrm{dns_name}))$$

Erişim sertifikası artık güncellenebilir bir alandır; eski ve yeni parmak izi bir olayla duyurulur, geçmiş kalır. Bir parmak izi aynı anda yalnız bir doğrulayıcıya ait olabilir. Alan adının kendisi deftere metin olarak yazılmaz; kontrat yalnız özeti görür.

9.2 İki sorun: kapma ve kalıcı kilit

Bu karar, bir güvenlik incelemesinde iki yeni soruyu ortaya çıkardı (ADR-0075) [12]. Kontrat doğrulayıcı kimliğini "ilk gelen alır" kuralıyla veriyordu ve alan adının hangi devlete ait olduğunu bilemezdi, çünkü elinde yalnız özet vardı.

Kapma. Bir devlet, başka bir ülkedeki bir kurumun alan adından türetilen kimliği önce kaydedebilirdi. Asıl sahibinin devleti o kurumu kaydetmeye çalıştığında kontrat "bu kimlik zaten var" diyerek reddederdi. Egemenlik kuralı yalnız "kendi devlet kodunla yaz" diyor; kimliğin hangi devlete düşeceğini söylemiyordu.

Kalıcı kilit. İptal kesindir. Bir doğrulayıcı bir kez iptal edilirse aynı kimlik bir daha hiç kaydedilemezdi. Alan adı el değiştirirse ya da iptal yanlışlıkla yapılmışsa, o alan adıyla ağda doğrulayıcı olmak imkânsız hâle gelirdi.

9.3 Çözüm: dönemler ve ortak oy

İki çözüm tartışıldı. Birincisi, kimliğin sahibini değiştirmeyi ve iptal edilmiş kimliği yeniden açmayı devletlerin ortak oyuna bırakmaktı. İkincisi, alan adı sahipliğini bir DNS kaydıyla zincir dışında kanıtlatıp sonra oya sunmaktı. Proje yönetimi birinciyi seçti; DNS kanıtı, oy veren devletlere sunulabilecek destekleyici bir belge olarak kaldı [12]. Gerekçe şudur: bir kimliğin hangi devlete düşeceği ağ düzeyinde bir karardır ve ağın ortak kapısı yönetişimdir; DNS de geçici olarak ele geçirilebilir ve tek başına yetmez.

Kurallar şöyle işler (RO1–RO5) [12]:

- Bir doğrulayıcı kimliğinin sahibi devlet yalnız türü belli bir **devir** önerisiyle, üçte iki oyla değişir.
- İptal bir dönem için kesindir. Aynı kimliğin **yeniden kaydı** yalnız türü belli bir yeniden kayıt önerisiyle, üçte iki oyla açılır.
- Devir ve yeniden kayıt yeni bir **kayıt dönemi** başlatır. Kayıt silinmez; durum geçmişi ve dönemler yalnız eklenir. Yeni dönem eski dönemin kapsamlarını, kayıt verisini ve kapı gruplarını devralmaz.
- Önerinin içeriği önce sabitlenir; sabitlemeden sonra kayda başka bir dönem eklendiyse öneri bayattır ve yazılmaz.
- İmzalı listede de aynı kural geçerlidir.

SOLIDITY Örnek: dönem değişikliğinin sabitlenen içeriği (SPEC-BC-0001 §6.1)

```
enum PeriodAction { NONE, REASSIGNMENT, RE_REGISTRATION }
struct PeriodChange {             // serbest metin yok; gerekçe yalnız özet
    PeriodAction action; bytes32 rpId; bytes2 fromState; bytes2 toState;
    bytes32 nameHash; bytes32 accessCertFingerprint; bytes32 reasonHash;
}
function stagePeriodChange(PeriodChange calldata change) external returns (bytes32 stageId); // herkes; yalnız
saklar
function commitPeriodChange(bytes32 stageId, PeriodAction action) external;               // yalnız Governance
function relyingPartyAt(bytes32 rpId, uint64 t) external view returns (RPStatus, bytes2 stateCode, uint32 period);
```

Son satırdaki okuma, Bölüm 8'deki ilkenin doğrulayıcı tarafındaki karşılığıdır: geçmiş bir an için soru sorulduğunda cevap o andaki döneme göre verilir.

Durum: Kontratlarda ve liste yayıncısının kayıt aracında yerel olarak **çalışıyor**. Aynı alan adının devirden sonra iki ülke listesinde birden kalması hâli için kural henüz paket değişikliği bekliyor; bugün liste kaynağı bu durumu belirsiz sayar ve kapalı güvenli davranır (Bölüm 10.4).

10 Listelerden Deftere Geçiş

10.1 Geçiş bir yeniden yazım değil

Defter bir gün açıldığında yeni bir başlangıç yapmayacak. Listelerin bugüne kadarki bütün geçmişi kontratlara bir kez aktarılacak ve defter, listeyle her soruda aynı cevabı verdiği kanıtlanmadan devreye girmeyecek. ADR-0009 bunu en başta yazmıştı: geçiş, liste sürüm arşivinin kontrat çağrılarına **yeniden oynatılması** ve bir **eşdeğerlik testidir** (K7) [3]. Verilmiş belgeler yeniden verilmez; cüzdan ve belge verme hizmeti değişmez; istemciler yalnız `TrustSource` arayüzünün arkasındaki uygulamayı değiştirir [15].

Neden geçmişi aktarmak gerekiyor? Çünkü defter açıldığı günden başlasaydı, Aynur'un bugün verilmiş diploması defterde hiç var olmayan bir kurumun belgesi gibi görünürdü. Kabul ölçütü de buna göre yazılır: kaydedilmiş her güven sorusunda listeye dayanan kaynak ile deftere dayanan kaynak aynı cevabı vermelidir; tek bir soruda kabul ile ret ayrışırsa geçiş tamamlanmamıştır.

10.2 Bugünkü kontratlar geçmişi neden yazamıyordu?

Kasım 2026'daki karşılaştırma bunun göründüğü kadar kolay olmadığını gösterdi [10]. Kontratlar tarih taşıyan her yazımda o anki blok zamanını kullanıyordu. Geçmiş, oynatıldığı günün tarihini alacaktı ve

eski belgeler reddedilecekti. Sert iptal yoktu. Bir belge türünün yalnız tek bir içerik özeti kaydedilebiliyordu, oysa listeler aynı türün birden çok geçerli sürümünü tanır. Doğrulayıcı kimliği sertifikaya bağlıydı. Bir de günlük kotalar vardı: aylarca biriken bir geçmişi kotalar altında yazmak haftalar sürer, belki hiç bitmezdi.

Bunların bir kısmı kontratlara yeni yetenekler eklenerek çözüldü (sert iptal, belge türü sürüm özetleri, alan adına bağlı kimlik, `iat` anındaki durum). Ama tarih sorunu daha incedir ve ayrı bir karar gerektirdi.

10.3 Tek seferlik göç penceresi

Kontratlar geçmiş tarih yazabilsin mi? Her zaman yazabilseydi tehlikeli olurdu: bir devletin delege anahtarı ele geçirilirse, saldırgan geçmişi sessizce yeniden yazabilirdi. Hiç yazamasaydı geçmiş aktarılamazdı. ADR-0073'ün cevabı bir **göç penceresidir** (K1, CE1) [10]:

- Her kayıt kontratı üç aşamalı bir göç durumu taşır: **başlamadı → açık → kapandı**. Geçişler yalnız ileri gider. "Kapandı"dan geri dönüş yoktur ve hiçbir yükseltme bu durumu sıfırlayamaz.
- Pencereyi devletlerin üçte iki oyuyla, türü belli bir öneri **bir kez** açar. Öneri, yeniden oynatma planının özetini (`replayPlanHash`) taşır. Plan herkese açık arşivden deterministik olarak üretilir; isteyen herkes planı arşivden yeniden üretip zincirdeki olaylarla karşılaştırabilir.
- Pencere **en çok 30 gündür**. Yalnız açıkken ve bu tavanı aşmadan uzatılabilir. Süre dolunca kendiliğinden kapanmış sayılır.
- Pencere açıkken tarih taşıyan yazımlar tarihi açıkça alır. Hiçbir tarih o anki blok zamanından ileri olamaz; bir kaydın geçmişi zaman sırasıyla yazılır; yazılmış geçmiş değiştirilmez.
- Pencere açıkken günlük kotalar uygulanmaz; kapanınca aynen döner.
- Ulusal kayıtları yine o devletin delegeleri yazar. Ağ düzeyi kayıtları açılış önerisinin yetkilendirdiği tek bir göç yürütücüsü yazar; her kayıt için ayrı oy gerekmez.
- Pencere açıkken tarihsiz olağan yazım dondurulur; acil bir yazım açık tarihle yapılır. Böylece yeniden oynatılan geçmiş ile o günkü işletim aynı kaydın geçmişinde sessizce yarışmaz.

TEKNİK AYRINTI

Göç penceresinin kuralları

t_0 pencerenin açıldığı an, t_{son} pencerenin son anı, τ bir yazımda verilen tarih ve b o yazımın girdiği bloğun zamanı olsun [10, 14]:

$$t_0 \le b \le t_{\text{son}} \le t_0 + 30 \text{ gün}$$

$$\tau \le b$$

$$\tau_{j+1} \ge \tau_j \quad \text{(aynı kayıt için)}$$

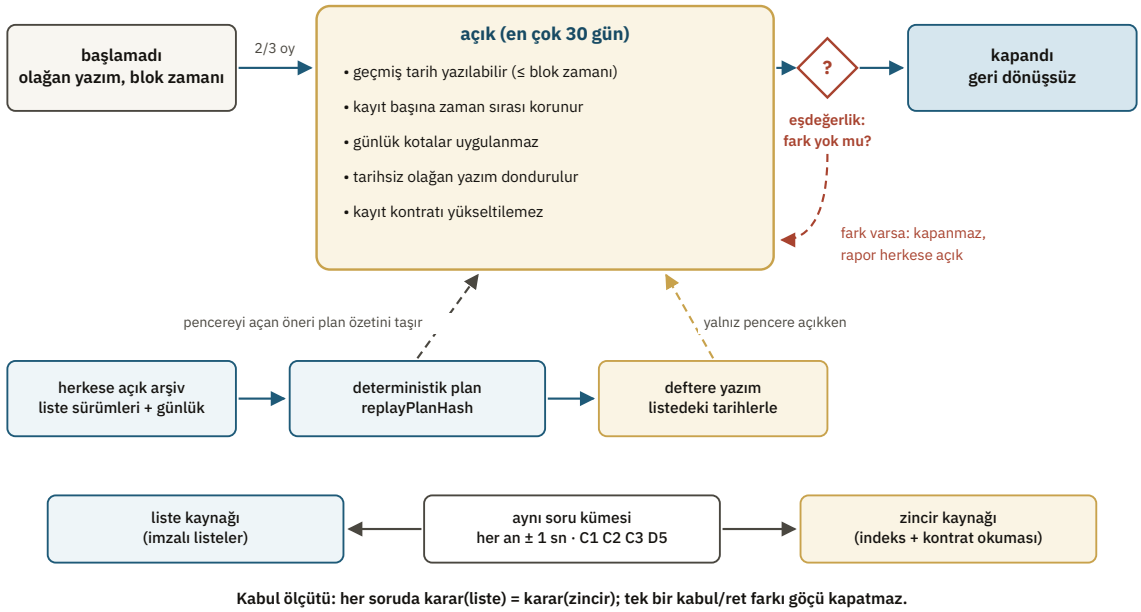
$$\text{faz} : \text{başlamadı} \to \text{açık} \to \text{kapandı} \quad \text{(geri dönüşsüz)}$$

Kontratlarda "yükseltmeyle bile yeniden açılamaz" kuralı üç katmanla korunur: pencere açıkken kayıt kontratı yükseltilemez; `Governance` her yükseltmede hedef kontratın göç durumunu önce ve sonra okur, fark varsa yükseltmeyi geri alır; göç durumunu tutan kaynak kodun özeti sabitlenir ve yükseltme incelemesi bu kaynağa başka hiçbir kodun dokunmadığını denetler. İkinci katman tek başına yetmez, çünkü yeni bir sürüm göç durumuna sonradan yazan bir fonksiyon taşıyabilir; bu yüzden sınır yükseltme incelemesine de dayanır.

Yeniden oynatma bir komutla yapılır (`trust:chain-replay`). Komut önce arşivi doğrular: her liste sürümünün imzası, sürüm zinciri, çapa günlüğü zinciri ve ortam. Sonra üretim arşivinin ilk arşivlenmiş sürümünden başlayan deterministik bir plan ve onun özetini üretir. Varsayılan davranış yalnız planı basmaktır. Yazım yalnız özellik bayrağı açıkken, pencere bu plan özetiyle açıkken ve gerçek ağda

ayrıca bir onay değişkeni plan özetiyle verildiğinde yapılır. Komut yarıda kesilirse kaldığı yerden sürer: zincirde olan işlemi atlar [14].

1	`Devletler (2/3)`	**Pencereyi aç.** Plan özetini taşıyan öneri oylanır; süre ve göç yürütücüsü belirlenir.	$\text{faz} = \text{açık}, \quad t_{\text{son}} \leq t_0 + 30\ \text{gün}$
2	`Yeniden oynatma`	**Arşivi doğrula.** İmzalar, sürüm zinciri, günlük zinciri ve ortam denetlenir.	$\text{replayPlanHash} = \text{SHA-256}(\text{plan})$
3	`Devlet delegeleri`	**Ulusal geçmişi yaz.** Kök, kurum, yetki ve durum değişiklikleri listedeki tarihleriyle, sırayla.	$\tau \leq b$
4	`Göç yürütücüsü`	**Ağ düzeyi geçmişi yaz.** Ortak belge türleri ve cüzdan sağlayıcıları.	$\tau_{j+1} \geq \tau_j$
5	`İndeksleyici`	**Okuma kaynağını kur.** Olaylardan dizin üretilir; `ChainTrustSource` hazır olur.	$\text{dizin} = \text{olaylar}(b)$
6	`Eşdeğerlik testi`	**Karşılaştır.** Aynı sorular iki kaynağa sorulur; her soruda karar aynı olmalı.	$\text{karar}_{\text{liste}} = \text{karar}_{\text{zincir}}$
7	`Devletler (2/3)`	**Pencereyi kapat.** Yalnız fark yoksa; rapor herkese açık.	$\text{faz} = \text{kapandı}$



Şekil 6. Geçiş: tek seferlik göç penceresi ve eşdeğerlik kapısı

10.4 Eşdeğerlik nasıl sınanır?

Eşdeğerlik testinin yöntemi sonucundan daha önemlidir; çünkü sonuç ancak yöntem kadar güvenilirdir. Test şöyle kurulur [14]:

1. Bir **arşiv** hazırlanır. Gerçek geçişte bu, üretim arşivinin kendisidir. Bugünkü testte birkaç sürümlük bir deneme arşividir ve bilerek zor durumları içerir: askıya alma ve yeniden etkinleşme, yumuşak ve sert iptal, emekliye ayrılan ve halefi olan bir kök, bir belge türünün birden çok sürümü, kullanımdan kaldırılan ve iptal edilen türler, belge türü yetki pencereleri, tanımanın açılıp kapanması, cüzdan sağlayıcı anahtarları, kullanım kapsamları, kapı grupları, iptalden sonra yeniden kayıt ve iki devlet arasında devir.
2. Arşiv kontratlara yeniden oynatılır ve indeksleyici çalıştırılır.

3. Bir **soru kümesi** üretilir. Her kurum için arşivde kayıtlı her an sorulur; buna ek olarak her durum değişikliğinin **sınır anları** sorulur: değişiklikten bir saniye önce, tam değişiklik anında ve bir saniye sonra. Hataların çoğu sınırlarda saklanır; Bölüm 8'deki bir saniyelik fark da böyle bulunmuştu.
4. Her soru **üç kaynağa** sorulur: imzalı listeye dayanan kaynak, indeksleyicinin dizininden cevap veren zincir kaynağı ve doğrudan kontrat okuması. Üçüncü kaynak, indeksleyicinin kendisinin bir hata saklamadığını gösterir.
5. Cevaplar C1, C2, C3 ve D5 sorularında karşılaştırılır. Kabul ölçütü tektir: kabul ile ret arasında hiçbir fark olmamalıdır.

İlk denemede bu yöntemle 493 soru soruldu ve C1, C2, C3 ve D5'te üç kaynak arasında hiçbir kabul ya da ret farkı çıkmadı [30]. Bir tane **belgelenmiş bilinen fark** vardır: bir devirden sonra aynı alan adı iki ülke listesinde birden kalırsa liste kaynağı bu kimliği belirsiz sayar ve kapalı güvenli davranır; zincirde ise tek kayıt vardır, yeni sahibinde. Bu farkın kapanması bir paket değişikliği gerektiriyor ve açık iş olarak duruyor. Test her kod değişikliğinde sürekli tümleştirme hattında yeniden çalışır.

Bu sayıyı bir ilk sonuç olarak okumak gerekir. Gerçek geçişte soru kümesi üretim arşivinden ve pilot boyunca kaydedilen sorgulardan türetilecektir. Ayrıca belgelenmiş birkaç sınır daha vardır: doğrulayıcı durum yazımları kontratta tarih taşımadığı için zincirdeki geçmiş, yeniden oynatma anını gösterir (bugünkü durum aynıdır, doğrulama soruları etkilenmez); çapa günlüğü iptal listesinin bit sayısını ve boyutunu taşımadığı için gerçek ağ yazımı bu bilgiyi iptal listelerinin kendisinden okumak zorundadır; ağ düzeyi kayıtların bir kısmı henüz plana girmemiştir [14].

> ÖNEMLİ
>
> **Kapanıştan sonra da projeksiyon**
>
> Göç penceresi kapandıktan sonra da defter, listelerin projeksiyonu olarak kalır (CE6, CH1). Listeler defter aşamasında da yayımlanmayı sürdürür. Defterin bir gün kaynak hâline gelip gelmeyeceği ayrı bir karardır ve bugün alınmış bir karar yoktur [10].

Durum: Yeniden oynatma komutu, indeksleyici ve eşdeğerlik testi yerel olarak **çalışıyor** test sürekli tümleştirme hattında koşuyor. Gerçek geçiş **planlanan** durumda ve Bölüm 12'deki önkoşullara bağlı.

11 Yönetişim

11.1 Üç katman

Tamga'nın yönetişimi egemenlik öncelikli üç katmana ayrılır (ADR-0002) [23]. Ağa üyelik, yani yeni bir devletin onaylayıcı olması ya da çıkarılması ve protokol yükseltmesi, devletlerin üçte iki oyuyla yapılır. Ulusal kayıtlar yalnız ilgili devletin yetkisindedir ve hiçbir dış oya tabi değildir. Sınır ötesi tanıma, yani bir devletin başka bir devletin kurumlarını tanıyıp tanımayacağı, her devletin kendi tek taraflı kararıdır.

Bugün yönetişim belgede yaşar; liste aşamasında oy kuralları askıdadır ve ağın tek üyesi vardır. Defterde ise bu kurallar kodun parçasıdır. Üçte iki kuralı küçük ağlarda dejenere olabileceği için bir alt sınır taşır: gereken oy sayısı en az ikidir. Bunun bir sonucu bilinçli bir anayasal korumadır: iki üyeli bir ağda bir devlet diğerini tek başına çıkaramaz; iki taraf da yalnız gönüllü olarak çekilebilir [14].

$$\text{gerekli}(n) = \max\left(\left\lceil \frac{2n}{3} \right\rceil, 2\right)$$

$$n = 2 \Longrightarrow \text{gerekli} = 2 \quad \text{(çıkarmada çıkarılanın oyu sayılmaz: imkânsız)}$$

Her devletin tek taraflı çekilme hakkı vardır ve bu hak oy gerektirmez. Bir devletin çekilmesi ya da çıkarılması mevcut kayıtları ve o devletin kurumlarının verdiği belgeleri geçersiz kılmaz; yalnız yeni yazımı ve blok üretimini durdurur (GV1) [14, 23].

11.2 Yükseltmede bekleme süresi

Ağın kurallarını değiştiren en güçlü işlem kontrat yükseltmesidir. Çekilme hakkı, yükseltme oy geçtiği anda yürürlüğe girerse pratikte işlemez: itiraz eden devlet karar veremeden yeni kurallar uygulanmış olur. ADR-0055 bu yüzden yükseltmeye bir bekleme koydu [9].

1	`Üye devlet → Defter`	**Öneri.** Yeni sürümü gösteren öneri açılır; serbest veri taşımaz.	$\text{öneri} = (\text{hedef}, \text{yeni sürüm})$
2	`Üye devletler`	**Oylama.** Etkin üyeler oy verir.	$e \geq \text{gerekli}(n)$
3	`Defter`	**Kuyruk.** Eşik dolunca öneri uygulanmaz, kuyruğa girer.	$\text{eta} = t_{\text{kuyruk}} + 7\ \text{gün}$
4	`Üye devletler`	**Çekilme penceresi.** Herkes ne olacağını görür; katılmayan devlet çekilebilir.	$t_{\text{kuyruk}} \leq t < \text{eta}$
5	`Defter`	**Yeniden sayım ve uygulama.** Eşik o anki etkin üyelerden yeniden sayılır; 30 gün içinde uygulanmazsa öneri düşer.	$\text{eta} \leq t_{\text{uygulama}} \leq \text{eta} + 30\ \text{gün}$

Yedi gün bir uzlaşmadır: iki gün devlet kurumlarına tepki süresi bırakmaz, on dört gün acil bir düzeltmeyi gereksiz uzatır [9]. Bekleme yalnız yükseltmeye uygulanır; üye kabulü, çıkarma ve ortak belge türü kaydı beklemesizdir.

Oylanan şey okunabilir olmalı. Yönetişimin bir kuralı da oylanan şeyin okunabilir olmasıdır. Yükseltmeyle yalnız parametresiz bir başlangıç fonksiyonu çağrılabilir; başlangıç değerleri yeni sürümün kodundadır ve oylanan şey o koddur. Yazıcı eklemek, eşiği değiştirmek, ortak bir belge türünü kullanımdan kaldırmak, doğrulayıcı kimliğini devretmek ya da göç penceresini açmak gibi işlemlerin her biri türü belli ayrı bir öneri fonksiyonuyla yapılır (GE3) [9]. Genel bir "şu çağrıyı yürüt" önerisi bilerek kurulmadı: defterde serbest veri alanı açar ve bir devletin neye oy verdiğini okumasını zorlaştırırdı. Bir öneren, önerisini yürütülmeden geri çekebilir [14].

Durum: Yönetişim kontratı ve bütün öneri türleri yerel geliştirme ağında **çalışıyor**. Yönetişimin kendisi, yani oy veren devletler, **planlanan**: bugün ağın tek üyesi vardır.

12 Bugün Nerede?

Timur'un ikinci sorusuna, "zincir ne zaman ve hangi kurallarla gelecek" sorusuna, artık bütün parçalarıyla bakabiliriz. Tablo 5 tasarımın her parçasının bugünkü durumunu gösteriyor.

Tablo 5. Zincir tasarımının parçaları ve bugünkü durumları (Kasım 2026).

Parça	Durum	Not
İmzalı listeler, sürüm zinciri, saatlik çapa günlüğü	Çalışıyor	Gerçek ağda canlı; güvenin bugünkü dayanağı
`iat` anındaki durum, sert iptal (listelerde)	Çalışıyor	Doğrulama paketinde
Kontratlar (dokuz kontrat ve okuma kapısı)	Çalışıyor	Derleniyor ve test ediliyor; üretimde yok
Yerel Besu geliştirme ağı (tek onaylayıcı, QBFT)	Çalışıyor	Yalnız yerel; kontratlar dağıtıldı, duman testi geçti
Liste çapası, eşikli yazıcılar, rakip adaylar	Çalışıyor	Yerelde; üretimde bayrak kapalı

Parça	Durum	Not
Yeniden oynatma, indeksleyici, eşdeğerlik testi	Çalışıyor	Yerelde ve sürekli tümleştirmede
Üretim defteri	Planlanan	İkinci bağımsız işletmeci ve bağımsız güvenlik denetimi önkoşulu
Göç penceresinin açılması ve gerçek geçiş	Planlanan	Üretim defterinden sonra; devletlerin oyu
Devletlerin oy verdiği yönetişim	Planlanan	Bugün tek üye
Cüzdanın zincir kaynağına geçişi	Planlanan	Aynı arayüzün arkasında, liste aynası ya da hafif indeksleyiciyle
Değer katmanı (ödeme, token)	Araştırma	Askıda; ağ token ihraç etmez, değer aktarımı işlemez

Üretim defterinin iki önkoşulu vardır. Birincisi **ikinci bağımsız işletmecidir**. Geçici işletmeciden hukuken ve operasyonel olarak bağımsız en az bir kurum onaylayıcı işletmeyi yazılı olarak kabul etmedikçe defter kurulmaz ve özellik bayrağı açılmaz; üretimde onay eşiği de en az iki olmak zorundadır [3, 5]. Bugün böyle bir işletmeci yoktur [1]. İkincisi **bağımsız bir güvenlik denetimidir**. Kontrat şartnamesi bunu açıkça bir üretim önkoşulu sayar ve denetlenecek başlıkları da sıralar: yükseltme deseninin kilitlenme riski, depolama düzeni, egemenlik denetiminin atlatılma yolları ve kontratlar arası çağrılardaki yeniden girme riski [14]. Kontrat güvenlik turları ağın kendi içinde yapıldı ve bulguları kararlara işlendi; bunlar bağımsız denetimin yerini tutmaz.

Bir şey daha kayda geçmeli. Değer katmanı, yani defterin ileride ödeme ya da varlık aktarımı taşıması fikri, ilk tasarımın bir parçasıydı. Bugün bağlayıcı değildir ve yalnız bir araştırma yönü olarak kayıtta durur. Tamga Network token ya da coin ihraç etmez; ödeme, mutabakat ya da değer aktarımı işlemez. Bu yönün yeniden açılması yönetişim kurumunun kararını, bağımsız bir güvenlik incelemesini ve yeni bir karar kaydını gerektirir [28].

Açıklık da tasarımın parçasıdır. Proje yönetiminin kararıyla kontratlar, defter ağının yapılandırma dosyaları, zincir aşamasının belgeleri ve okuma tarafının kodu ağın herkese açık deposunda yer alır (ADR-0074 K6) [11]. Bu yazı da aynı kararın bir sonucudur.

13 Sınırlar ve Açık Sorular

İkinci işletmeci nereden gelecek? Tasarımın bütün değeri bağımsız ikinci bir tarafa bağlıdır ve bu taraf bugün yoktur. Kod hazır olsa da defter, onu birlikte işletecek kurumlar bulunana kadar yalnız bir hazırlıktır. Bu, tasarımın bir eksiği değil, dürüstlüğünün gereğidir; ama sonucunu değiştirmez: o güne kadar çift konuşmaya karşı koruma caydırma düzeyinde kalır.

Eşik ne olmalı? Üretimde onay eşiğinin en az iki olacağı kesindir; tam değeri ve ikinci yazıcının kim olacağı ikinci işletmeci adayıyla birlikte belirlenecek [5]. İki işletmecili bir defterde biri susarsa hiçbir sürüm onaylanamaz; bu da doğrulamayı etkilemez (cevap “bilinmiyor” olur), ama tanıklığı durdurur.

İzin listelerinin tutarlılığı işletmecilere kalır. Besu yerel izin kurallarını başka düğümün ürettiği bloklara uygulamadığı için listelerin aynı tutulması bir işletme disiplinidir. Araç farkı raporlar; ama farkı gidermek insanların işidir [6]. Devlet sayısı büyürse kendi izin kurallarını taşıyan bir Besu eklentisi yeniden değerlendirilebilir.

Özet her şeyi gizlemez. Kurum adları ve alan adları sonlu bir kümedir; bir özetten geriye doğru tahminle adı bulmak mümkündür. Kontrat şartnamesi bunu açıkça yazar: ad özeti gerçek bir koruma değil, bir sürtünmedir [14]. Bu bilgiler zaten kamuya açık olduğundan sorun değildir; ama aynı yaklaşımın kişisel veriye uygulanamayacağının nedeni de budur. Kişisel veri özet olarak bile yazılmaz.

Defter de bir gözlem yüzeyidir. İzinli ağda her işlem her onaylayıcıya görünür. İptal listelerinin yayını sabit aralıklarla yapılmazsa, hangi kurumun ne zaman iptal yaptığı yayın zamanlamasından sezilebilir. Tasarım bu yüzden saatlik ve düzenli yayın ister [14].

Bilinen farkın kapanması. Devirden sonra iki listede kalan alan adı için önerilen kural (eski listedeki kaydı gerekçesiyle iptal etmek ve doğrulama kodunun bu durumu belirsiz saymaması) bir paket değişikliği istiyor ve henüz uygulanmadı [14].

Hukuki tanıma ayrı bir yoldur. Defter ne kadar iyi tasarlanırsa tasarlansın, bir belgenin başka bir ülkede hukuken geçerli sayılması devletler arası anlaşmalara bağlıdır. Ağın rolü teknik hazırlıktır [1].

14 Sonuç

Tamga Network'ün zincir tasarımı bir bekleyişin tasarımıdır, ama boş bir bekleyişin değil. Defter bugün kapalı; buna karşılık kodu yazılmış, kuralları karara bağlanmış ve listelerle aynı cevabı verdiği sınanmıştır. Bekleyişin nedeni teknik değil yapısaldır: bir defter ancak onu birlikte tutan bağımsız taraflar kadar değerlidir ve o taraflar henüz yoktur.

Tasarımın omurgası tek bir yön kuralıdır. Kaynak imzalı listelerdir; defter onların projeksiyonudur. Bu kuraldan art arda başka kurallar çıkar. Doğrulama deftere bağlı olamaz, çünkü kaynak listedir. Deftere kişisel veri yazılamaz, çünkü defterin işi kişileri değil kurumları ve listeleri tanımaktır. Geçmiş deftere yalnız bir kez ve denetimli bir pencerede yazılabilir, çünkü geçmişi yazabilen bir anahtar onu değiştirebilen bir anahtardır. Geçiş bir eşdeğerlik testine bağlıdır, çünkü ölçü Aynur'un diplomasıdır: bugün geçerli olan, yarın da aynı cevabı almalıdır.

Timur'un iki sorusunun cevabı da buradadır. Tek bir işletmecinin çift konuşmasına karşı bugünkü koruma caydırmadır; defter onu uzlaşmayla engellenen bir şeye çevirebilir, ama ancak Timur'un kurumu gibi bağımsız bir taraf aynı özeti kendi başına hesaplayıp gönderdiğinde. Zincirin ne zaman geleceği sorusunun cevabı da bir tarih olarak verilemez. Cevap bir koşuldur: ikinci bağımsız işletmeci ve bağımsız bir güvenlik denetimi. Koşul sağlandığında geçişin nasıl yapılacağı ise bugünden yazılıdır.

Kaynaklar

[1] Tamga Network (2026). *Türkiye ve Türk Dünyası için Dijital Güven Altyapısı.* Whitepaper, sürüm 1.0, Bölüm 6 ve 19. https://tamga.network/whitepaper-tr.pdf

[2] Tamga Network Akademik Çalışmalar, A12. *Blokzincir ve Dijital Kimlik: Ne Zaman Gerekir, Ne Zaman Gerekmez?* Sürüm 1.0. tamga.network/publications

[3] Tamga Network. *ADR-0009: Zincirsiz beta ve zincir eşiği.* Karar kaydı, 24 Eylül 2026. https://docs.tamga.network/tr/adr/0009-phase-b-chainless-beta-and-chain-threshold

[4] Tamga Network. *ADR-0001: Besu QBFT defteri.* Karar kaydı, 29 Temmuz 2026. https://docs.tamga.network/tr/adr/0001-blockchain-framework-besu-qbft

[5] Tamga Network. *ADR-0051: Zincir, imzalı listelerin projeksiyonudur.* Karar kaydı, kabul 19 Ekim 2026. https://docs.tamga.network/tr/adr/0051-chain-as-projection-of-signed-lists

[6] Tamga Network. *ADR-0052: Besu yerel izinlendirme.* Karar kaydı, 19 Ekim 2026. https://docs.tamga.network/tr/adr/0052-besu-local-permissioning

[7] Tamga Network. *ADR-0053: Liste çapasında rakip adaylar.* Karar kaydı, 19 Ekim 2026. https://docs.tamga.network/tr/adr/0053-anchor-competing-candidates

[8] Tamga Network. *ADR-0054: Zincirde şema ad alanı, devletler arası bağ ve kota kapsamı.* Karar kaydı, 19 Ekim 2026. https://docs.tamga.network/tr/adr/0054-chain-namespaces-references-quotas

[9] Tamga Network. *ADR-0055: Yönetişim: yükseltme bekleme süresi ve kayıt yönetimi önerileri.* Karar kaydı, 19 Ekim 2026. https://docs.tamga.network/tr/adr/0055-governance-upgrade-delay-and-registry-admin
[10] Tamga Network. *ADR-0073: Zincir–liste eşdeğerliğinin uygulanabilirliği: göç kipi, sert iptal, şema sürümleri, doğrulayıcı kimliği, durum eşlemesi.* Karar kaydı, 4 Kasım 2026. https://docs.tamga.network/tr/adr/0073-chain-list-equivalence-feasibility
[11] Tamga Network. *ADR-0074: Projeksiyon kapsamı 0.8: imzalı listelerdeki bütün güven kayıtları zincirde.* Karar kaydı, 4 Kasım 2026. https://docs.tamga.network/tr/adr/0074-chain-projection-scope
[12] Tamga Network. *ADR-0075: Doğrulayıcı alan adı kimliğinin sahipliği ve iptal sonrası yeniden kayıt.* Karar kaydı, 4 Kasım 2026. https://docs.tamga.network/tr/adr/0075-relying-party-domain-ownership
[13] Tamga Network. *SPEC-TRUST-0001: Güven listeleri.* Şartname, sürüm 1.0.0. https://docs.tamga.network/tr/specifications/trust-lists
[14] Tamga Network. *SPEC-BC-0001: Güven katmanı kontratları.* Şartname, sürüm 1.0.0 (§0, §1, §6, §9, §11, §12, §15, §16). Herkese açık depo: github.com/tamga-network/tamga-network, `docs/ledger/specifications/0001-trust-layer-contracts.md`.
[15] Tamga Network. *ADR-0015: Tek güven arayüzü.* Karar kaydı, 27 Eylül 2026. https://docs.tamga.network/tr/adr/0015-single-trust-interface-for-clients
[16] ETSI (2025). *ETSI TS 119 612 V2.4.1: Electronic Signatures and Trust Infrastructures (ESI); Trusted Lists.* Ağustos 2025.
[17] Haber, S. ve Stornetta, W. S. (1991). "How to Time-Stamp a Digital Document." *Journal of Cryptology*, 3(2), 99–111.
[18] Saltini, R. ve Hyland-Wood, D. (2019). "IBFT 2.0: A Safe and Live Variation of the IBFT Blockchain Consensus Protocol for Eventually Synchronous Networks." arXiv:1909.10194.
[19] Besu belgeleri. "QBFT" ve "Proof of authority consensus." besu.hyperledger.org (erişim: Kasım 2026).
[20] Lamport, L., Shostak, R. ve Pease, M. (1982). "The Byzantine Generals Problem." *ACM Transactions on Programming Languages and Systems*, 4(3), 382–401.
[21] Hyperledger Besu sürüm notları, 24.12.0 ve 25.6.0 ("Deprecations", "Sunset features"). github.com/besu-eth/besu/releases (erişim: Kasım 2026).
[22] Laurie, B., Messeri, E. ve Stradling, R. (2021). *Certificate Transparency Version 2.0.* IETF RFC 9162.
[23] Tamga Network. *ADR-0002: Egemenlik öncelikli yönetişim.* Karar kaydı, 5 Ağustos 2026. https://docs.tamga.network/tr/adr/0002-sovereignty-first-governance
[24] Ethereum Improvement Proposals. *ERC-1822: Universal Upgradeable Proxy Standard (UUPS) ERC-1967: Proxy Storage Slots ERC-7201: Namespaced Storage Layout.* eips.ethereum.org (erişim: Kasım 2026).
[25] Looker, T., Bastian, P. ve Bormann, C. *Token Status List (TSL).* IETF OAuth Çalışma Grubu İnternet taslağı, draft-ietf-oauth-status-list (yazının yayın tarihindeki son sürüm).
[26] Wood, G. *Ethereum: A Secure Decentralised Generalised Transaction Ledger* ("Yellow Paper"), sürekli güncellenen sürüm. ethereum.github.io/yellowpaper. Ayrıca: NIST (2015). *FIPS 202: SHA-3 Standard.* ve NIST (2015). *FIPS 180-4: Secure Hash Standard.*
[27] Rundgren, A., Jordan, B. ve Erdtman, S. (2020). *JSON Canonicalization Scheme (JCS).* IETF RFC 8785.
[28] Tamga Network. *ADR-0072: Değer katmanı askıda.* Karar kaydı, 30 Ekim 2026. https://docs.tamga.network/tr/adr/0072-value-layer-suspended
[29] Tamga Network. *ARCH-0002: Besu ağı kurulumu* ve *ARCH-0006: Canlıya alma adımları.* Mimari belgeleri. Herkese açık depo: github.com/tamga-network/tamga-network, `docs/ledger/architecture/`.

[30] Tamga Network. *Değişiklik günlüğü (CHANGELOG)*, “chain: reading and transfer side” maddesi. Herkese açık depo: github.com/tamga-network/tamga-network, `CHANGELOG.md`.

Ek A — Sözlük

Bu yazıda geçen terimler, yazıdaki anlamlarıyla. İngilizce karşılıklar başka kaynaklarda aranabilmeleri için verilmiştir.

Tablo A.1. Terimler sözlüğü.

Terim	İngilizce	Anlamı
Aday	candidate	Bir liste sürümü için yazıcıların gönderdiği, henüz onaylanmamış özet; farklı her özet ayrı adaydır
Ayrıntı özeti	details hash	Serbest metin ya da kayıt ayrıntısı bloğunun kanonik JSON’unun SHA-256’sı; metnin kendisi deftere yazılmaz
Çapa günlüğü	anchor log	İptal listesi yayınlarını ve belge türü değişikliklerini en az saatte bir kaydeden, satırları birbirine özetle bağlı imzalı günlük
Çift konuşma	split view, equivocation	Bir yayıncının farklı taraflara aynı anda farklı listeler göstermesi
Eşdeğerlik testi	equivalence test	Aynı soru kümesinin liste kaynağına ve zincir kaynağına sorulup kabul/ret cevaplarının karşılaştırılması
Eşik	threshold	Bir çapanın onaylı sayılması için aynı özeti desteklemesi gereken farklı yetkili yazıcı sayısı
Göç penceresi	migration window	Geçmişin deftere açık tarihlerle yazılabildiği, en çok 30 günlük, bir kez açılan ve geri dönüşsüz kapanan dönem
İndeksleyici	indexer	Kontrat olaylarını izleyip kayıtları yerel bir dizine yazan, doğrulamanın zincire doğrudan sormamasını sağlayan bileşen
İzinli defter	permissioned ledger	Yalnız onaylı düğümlerin bağlandığı ve onaylı hesapların işlem gönderdiği defter
Kayıt dönemi	registration period	Bir doğrulayıcı kimliğinin bir devlete ait olduğu, devir ya da yeniden kayıtla yenisi eklenen dönem
Kontrat	smart contract	Defterde saklanan ve her onaylayıcının aynı biçimde çalıştırdığı program; hukuki sözleşme değildir
Onaylayıcı	validator	Blokları onaylayan, kimliği bilinen katılımcı; tasarımda devletler
Projeksiyon	projection	Defterin imzalı listelerin ikinci, türev kaydı olması; çelişkide liste geçerlidir
QBFT	QBFT	Besu’nun Bizans hatasına dayanıklı, yetki kanıtına dayanan, blokları tek adımda kesinleştiren uzlaşma yöntemi
Sert iptal	hard revocation	Bir kurumun belirli bir andan sonra verdiği bütün belgeleri düşüren iptal (`invalidates_from`)
Yarı açık aralık	half-open interval	Başlangıcı dahil, bitişi hariç zaman aralığı: [başlangıç, bitiş)
Yazıcı	anchorer	Liste özetlerini deftere gönderme yetkisi olan işletmeci hesabı; her gönderim bir oydur
Yeniden oynatma	replay	Liste sürüm arşivinin sırayla kontrat çağrılarına dönüştürülüp deftere yazılması
Yumuşak iptal	soft revocation	Kurumun bundan sonra belge vermemesi; önceki belgeleri geçerli kalır
`iat`	issued at	Belgenin verildiği an; doğrulama kurumun durumuna bu anda bakar