

# Aelf Ekonomik ve Yönetişim Whitepaper'ı



V1.0

17 Şubat 2020

# İÇERİK

<b>1. Genel Bakış</b>	<b>4</b>
1.1 Ekonomik Model Genel Bakış	4
1.2 Sistemdeki Rollerin Tanımı	4
<b>2. Token Modeli</b>	<b>4</b>
2.1 Token Türleri	4
2.2 İhraç/Çıkarma ve Piyasaya Sürme Planı	5
2.3 Yakma Mekanizması	5
2.4 Kaynak Tokeni Ticaret Modeli	5
2.5 Zaman Başına Ödeme Modeli	7
2.6 Çapraz Zincir İndeks Ücreti Modeli	7
2.7 Gelir Paylaşımı Modeli	8
2.8 İşlem Ücreti Modeli	8
2.8.1 Geliştirici Ücretlendirme Modeli	8
2.8.2 Kullanıcı Ücretlendirme Modeli	11
<b>3. Teşvik Modeli - Düğümler ve Seçmenler</b>	<b>12</b>
3.1 Ödül Havuzu	12
3.2 Üretim Düğümleri Teşvikleri	13
3.3 Aday Düğüm Teşvikleri	14
3.4 Seçmen Teşvikleri	14
3.5 Düğüm Sayısı	15
3.6 Üretim Düğümü Seçimleri	15
3.6.1 Seçim Gereksinimleri	15
3.6.2 Seçim Süreci	16
3.7 Oylama Kuralları	16
3.8 Üretim Düğümü Eliminasyon Mekanizması	16
3.8.1 Seçim Dönemi	16
3.8.2 Eliminasyon Kuralı	17
3.9 Token Sahibi Sözleşmesinde Ödüller	17

<b>4. Teşvik Modeli - Yan Zincir Geliştiricileri</b>	<b>17</b>
4.1 Yan Zincir Oluşturma	17
4.2 Yan Zincir Kapatma Mekanizması	18
4.3 Yan Zincir Geliştirici Ücretlendirme Modeli	18
4.3.1 Özel Yan Zincir Ücretlendirme Modeli	18
4.3.2 Yan Zincir Destek Planı Ücretlendirme Modeli	19
4.3.3 Paylaşım Yan Zinciri Ücretlendirme Modeli	19
4.4 Üretim Düğümleri Teşvikleri	20
4.4.1 Yan Zincirlerde Sorumluluklar	20
4.4.2 Yan Zincir Ekosistemlerinde Ödüller	20
<b>5. Yönetişim</b>	<b>20</b>
5.1 Yönetişim modelleri	21
5.1.1 Parlamento Yönetişim Modeli	21
5.1.2 Birlik/Ortaklık Yönetişim Modeli	21
5.1.3 Referandum Yönetişim Modeli	22
5.1.4 Özelleştirilmiş Yönetim Modeli	23
5.2 Mevcut Ana Zincir Yönetişimi	24
5.2.1 Parlamento Yönetişim Modelini Kullanma	24
5.2.2 Birlik/Ortaklık Yönetişim Modelini Kullanma	25
5.2.3 Referandum Yönetişim Modelini Kullanma	25
5.2.4 Özel Bir Yönetişim Modeli Kullanma	25
5.3 Yan Zincir Yönetişimi	26
5.3.1 Parlamento Yönetişim Modelini Kullanma	27
5.3.2 Birlik/Ortaklık Yönetişim Modelini Kullanma	27
5.3.3 Özelleştirilmiş Yönetişim Modeli Kullanma	27

# 1. Genel Bakış

## 1.1 Ekonomik Model Genel Bakış

Bu belgede üretim düğümleri, aday düğümler, geliştiriciler ve token sahipleri için teşvikleri, hakları ve sorumlulukları ana hatlarıyla belirten Aelf ekosistemindeki çeşitli roller açıklanmaktadır.

Aelf ekonomik modeli; temel ekosistemin güçlendirilmesinde, ağın genel istikrarının ve işleyişinin sağlanmasında ve sürdürülebilir ve sağlıklı kalkınmayı teşvik etmek için topluluk iş birliğini teşvik etmede önemli bir rol oynamaktadır.

## 1.2 Sistemdeki Rollerin Tanımı

Aelf'in ekosisteminde şu roller vardır: üretim düğümleri, aday düğümler, geliştiriciler ve token sahipleri (kullanıcılar).

- **Üretim düğümü:** Herhangi bir kişi/organizasyon üretim düğümü olmak için başvurabilir. Başarılı olursa, konsensüs kurallarına göre blok üretimine katılacaklar ve blok ödülleri alacaklar.
- **Aday düğüm:** Oy sıralamasına göre bir üretim düğümü olmayan herhangi bir kişi/organizasyon.
- **Geliştirici:** Aelf ağında uygulamalar geliştiren bir kişi veya organizasyon.
- **Token sahibi (kullanıcı):** Aelf ağında herhangi bir Aelf platformunun değerli varlıklarına sahip olan ve geliştiriciler tarafından sunulan hizmetleri kullanmayı planlayan bir kişi veya organizasyon.

# 2. Token Modeli

## 2.1 Token Türleri

Üç tip token vardır: ELF token, kaynak tokenleri (CPU kaynakları, RAM kaynakları, DISK kaynakları, NET kaynakları, OKUMA kaynakları, YAZMA kaynakları, DEPOLAMA kaynakları, TRAFİK kaynakları) ve geliştiriciler tarafından oluşturulan tokenler;

- **ELF:** Aelf platformundaki ana token; işlem ücretleri, yan zincir indeks ücretleri, üretim düğümleri depozitoları, oylama ve blok ödülleri için kullanılır.
- **Kaynak Tokeni:** Bu token, geliştiriciler tarafından bir uygulama çalışırken kaynakların tüketimini ödemek için kullanılır. Geliştiricilerin, uygulamalarının normal gereksinimleri için yeterli kaynak tokeni olduğundan emin olmaları gerekir. Kaynak tokenleri, ELF tokenleriyle alınabilir ve satılabilir.
- **Geliştiriciler tarafından oluşturulan tokenler:** Aelf platformunda geliştiriciler, tokenler oluşturabilir ve kendi token modellerini ve teşvik mekanizmalarını kurabilirler.

## 2.2 İhraç/Çıkarma ve Piyasaya Sürme Planı

- Toplam ELF ihracı: 1 milyar
- Madencilik oranı: 120 milyon token (%12)
- Blok Ödülü Piyasaya Sürme Planı: her dört yılda bir yarıya indirilir
- Toplam ELF arzı ve kaynak tokenleri sabittir
- Toplam kaynak tokenleri ihracı: Her biri 500 milyon CPU kaynak tokenleri, RAM kaynak tokenleri, DISK kaynak tokenleri, NET kaynak tokenleri, OKUMA kaynak tokenleri, YAZMA kaynak tokenleri, DEPOLAMA kaynak tokenleri ve TRAFİK kaynak tokenleri.

## 2.3 Yakma Mekanizması

Aşağıdaki iki senaryo belirli bir oranda ELF'in yok edilmesi ile sonuçlanacaktır:

- **İşlem ücreti:** %10 imha edilecek ve %90 ödül havuzuna girecek
- **Kaynak satın alma:** Kaynak tokenleri satın almanın %0,5 ücret masrafı vardır ve bunların %50'si yok edilir ve %50'si ödül havuzuna girer

## 2.4 Kaynak Tokeni Ticaret Modeli

Kaynak token işlemleri, Bancor modelini kullanıyor. Alış ve satış fiyatları Bancor modeline tarafından belirlendiği için soruşturma süreci yoktur. Kullanıcılara, az sayıda kullanıcı ve küçük işlem hacmi ile bile verimli ve hızlı bir token ticaretini garanti edebilir; kaynak tokenlerinin ve ELF'in anında değişimini garanti edilebilir.

Kaynak tokenleri çıkarılırken toplam fiyat, sabit bir ELF fiyatına bağlanır:

$$cw \text{ (konektör ağırlığı)} = \frac{\text{konektör dengesi (bakiyesi)}}{\text{kaynak tokeni arzı} \times \text{fiyat}}$$

$cw$ , sabit bir değerdir; konektör dengesi, bağlanan ELF'in toplam miktarıdır. Bir miktar  $dS$  kaynak tokenleri satın alırken ve satarken, varsayalım:  $cw$ ,  $F$ 'dir; konektör dengesi,  $B$ 'dir; kaynak tokeni arzı,  $S$ 'dir; kaynak tokeni fiyatı,  $P$ 'dir, bu aşağıdaki denklemi verecektir:

$$B = FSP \quad (1)$$

$$Pd(S) = d(B) = d(FSP) = F(Pd(S) + Sd(P)) \quad (2)$$

$$\frac{dP}{dS} = \frac{1 - F}{F} \times \frac{P}{S} \quad (3)$$

$$\ln(P) = \frac{1 - F}{F} \times \ln(S) + C(\text{constant}) \quad (4)$$

Her zaman P ve S arasındaki ilişki belirlenebilir. Bu nedenle, herhangi bir zamanda, ELF harcayarak/alarak dS kaynak tokenleri satın alma/satma  $\int P dS$  ile hesaplanabilir.

Örneğin;  $\Delta S$  kaynak tokenleri satın alırken, aşağıdaki miktar kadar ELF'e ihtiyacınız vardır:

$$= \int_S^{S + \Delta S} P dS = FP_0 S_0 \left( \left( 1 + \frac{\Delta S}{S_0} \right)^{\frac{1}{F}} - 1 \right) = B_0 \left( \left( 1 + \frac{\Delta S}{S_0} \right)^{\frac{1}{F}} - 1 \right) \quad (5)$$

Aynı zamanda formül (5)'in ters işlemi, belirli bir miktarda ELF için kaç adet kaynak tokeninin değiştirilebileceğini elde etmek için kullanılabilir:

$$\Delta S = S_0 \left( \left( 1 + \frac{\Delta B}{B_0} \right)^F - 1 \right) \quad (6)$$

Relay tokeni gibi bir ara token varsa hem ELF tokenlerini hem de kaynak tokenlerini sabitler/bağlar. Bir kullanıcı belirli sayıda  $\Delta S$  Relay tokeni satın almak için  $\Delta B$  ELF tokeninin bir miktarını kullanmak isterse, birkaç kaynak tokeni satın almak için bu  $\Delta S$  Relay tokenleri kullanılır. Kaynak tokeni ve ELF token arasındaki ilişki:

Harcanan (elde edilen) ELF aşağıdaki formülle hesaplanır (a, kullanıcıların token satın alması/satması için bir değişkendir):

Satın Alma:

$$\text{cost} = bf \times \left( \left( \frac{bt}{bt - a} \right)^{\frac{wt}{wf}} - 1 \right) \quad (7)$$

Satma:

$$\text{gain} = bt \times \left( 1 - \left( \frac{bf}{bf + a} \right)^{\frac{wf}{wt}} \right) \quad (8)$$

(Cost = Maliyet, Gain = Kazanç)

Aşağıdaki sabitler, parlamento oyu ile belirlenir:

- bt= konektöre denge (konektörünün gerçek zamanlı konumu, başlangıçta üretim düğümleri tarafından belirlenir)
- bf= konektörden denge (konektörünün gerçek zamanlı konumu, başlangıçta üretim düğümleri tarafından belirlenir)
- wt= konektöre ağırlık (ağırlık sabit)
- wf= konektörden ağırlık (ağırlık sabit)

a, bir kullanıcı bir token satın aldığı/sattığında geçirilir; bu da satın alınan/satılan token sayısını gösterir. Satın alırken a değeri ne kadar büyük olursa, token başına daha fazla ELF harcanır. Satarken a değeri ne kadar büyük olursa, birim token fiyatı o kadar düşük olur.

Bu işlem modelini kullanarak ekonomik model açısından her iki konektörün değeri, herhangi bir zamanda Relay tokeninin eşdeğer miktarıdır.

Kaynak Token konektörü bakiyesinin başlangıçta A'ya ve ELF token konektörü bakiyesinin başlangıçta M'ye ayarlandığını varsayın. Belirli sayıda kaynak tokeni satın almak için harcanan N ELF tokeni sayısı bir ipotek hesabına konulduğunda:

- Kaynak Tokeni konektör dengesi ( $A - a$ )
- ELF tokeni konektör dengesi ( $M + N$ )

Modele göre herhangi bir anda iki tokenin değeri aynıdır,  $V_{A - a} = V_{M + N}$ , a son derece A'ya yakın olduğunda, geriye kalan kaynak tokenlerine karşılık gelen ELF fiyatının sonsuza yaklaştığı görülebilir. Bu denklem tutarlı bir şekilde tutulduğundan ne kadar çok kaynak tokeni satın alınırsa, o kadar çok ELF tokeni gerekecektir. Aksine, kullanıcılar daha fazla kaynak tokeni sattıklarında, daha fazla ELF tokeni elde ederler. Bu özellik, spekülasyondan kaçınmaya da yardımcı olacaktır.

## 2.5 Zaman Başına Ödeme Modeli

İnternet uygulaması geliştirme sürecinde geliştiriciler, genellikle uygulamalarını dağıtmak için sunucuları olarak bulut hizmeti kiralar ve daha sonra kullanıcılara hizmet sağlar. Uygulama, sadece sunucunun bilişim kaynaklarını kullanabilir. Aelf'de yan zincir geliştiricileri; CPU Token, RAM Token, DISK Token ve NET Token ödeyerek tek bir yan zincir kullanarak faydalanabilirler. Bu yan zincir, bağımsız bilişim ve depolama kaynaklarına sahiptir. Özel yan zincir, kullanılan zamana göre fiyatlandırılır. Ücretin birim fiyatı, üretim düğümü ile geliştirici arasındaki müzakere yoluyla belirlenir.

## 2.6 Çapraz Zincir İndeks Ücreti Modeli

Çapraz zincir transferleri ve çapraz zincir doğrulaması uygulamak isteyen yan zincir geliştiricilerinin, ana zincir indeksi yan zincirin blok bilgilerine sahip olması gerekir. Bunu kullanmak için blok indeks ücretini ödemeleri gerekir. Blok indeks ücreti için kullanılan birim ELF'dir. Alınan tutar, organizasyon ve geliştirici tarafından birlikte belirlenir. Bir yan zincir oluşturmak için başvuru yapılırken ilk indeks ücreti bir parametre olarak aktarılır. İndeks ücret tutarı, bir teklif aracılığıyla ayarlanabilir. Hem organizasyon hem de geliştirici düzeltilmiş planı kabul ettiğinde geçerli olacaktır.

## 2.7 Gelir Paylaşımı Modeli

Geliştiricilerin yan zincirleri kullanma eşliğini azaltmak için geliştiriciler, yan zincir kaynaklarını gelir paylaşımı biçiminde kullanmayı seçebilirler. Geliştirici, gelirin belirli bir yüzdesini üretim düğümlerine tahsis etmeyi vaat ediyor ve gelir payı her bir üretim düğümüne eşit olarak dağıtılacaktır. Geliştirici ve kuruluş, pay oranını ancak üzerinde anlaşmaya varıldıktan sonra ayarlanabilen veya güncellenebilen bir teklif biçiminde sürdürür. Geliştirici paylaşım modeli genellikle "destek planı" yan zincir türünde görünür (ayrıntılar için 4.3'e bakınız).

## 2.8 İşlem Ücreti Modeli

AElf sisteminde iki tür ücretlendirme modeli vardır: Geliştirici Ücretlendirme Modeli ve Kullanıcı Ücretlendirme Modeli. Sözleşmeler geliştirirken geliştiriciler, Geliştirici Ücretlendirme Modelini veya Kullanıcı Ücretlendirme Modelini seçebilir.

### 2.8.1 Geliştirici Ücretlendirme Modeli

Geliştirici Ücretlendirme Modeli, geliştiriciye işlemlerin yürütülmesine ilişkin ücretleri karşılaması için atar. Geliştirici Zamana Göre Öde modelinin aksine Geliştirici Ücretlendirme Modeli, isteğe bağlı bir ödeme yöntemidir. Bu model, işlemin yürütülmesi sırasında kaynakların kullanımını analiz ederek geliştirici işlem ücretlerini toplar. Kaynak tokenleri, merkezi olmayan uygulamalar (Dapps) tarafından kaynakların isteğe bağlı kullanımını ölçmek için kullanılır. Bu kaynak tokenleri şunları içerir: OKUMA tokeni, YAZMA tokeni, DEPOLAMA tokeni ve TRAFİK tokeni. Bunlar arasında, OKUMA/YAZMA tokenleri, işlem yürütüldüğünde durum veri tabanının okunması ve yazılması için kullanılır (CPU ve RAM isteğe bağlı olarak kullanılır). İşlem boyutu, ağ iletimini ve son depolamayı etkiler; DEPOLAMA tokeni istek üzerine kullanılan depolama kaynaklarını ödemek için kullanılır ve TRAFİK tokeni isteğe bağlı olarak kullanılması da dahil olmak üzere ağ bant genişliği kaynaklarını ödemek için kullanılır.

Her kaynağa, işlemlerde kullanımına göre bir eşik değeri verilir. Bu eşik, normal ücretlendirme aralığının yanı sıra eşik normal sınırdan daha fazla veya daha az olduğunda ayarlanmış ücretlendirme aralığını belirlemek için kullanılır. Normal ücretlendirme aralığında ücretler, kullanılan kaynağın birim fiyatına dayanır. Ekstra ücretlendirme aralıklarına ek olarak ilave güç tüketimi, fazlalığın kare seviyesine göre hesaplanır. Spesifik eşiklerin oluşturulması, fiili/gerçek işlemlerde çeşitli kaynakların kullanımına dayalı olarak hesaplanır.

- OKUMA (READ) Tokeni:

$$\text{token cost} = C + k_1 \times \text{count} \quad \text{count} \leq 10$$

$$\text{token cost} = k_2 \times \text{count} \quad 10 < \text{count} \leq 100$$

$$\text{token cost} = k_2 \times \text{count} + \left( \frac{\text{count}}{\text{changeSpan}} \right)^2 \times \frac{\text{weight}}{\text{weightBase}} \quad \text{count} > 100$$

**Count (Sayım):** tüketilen OKUMA kaynaklarını ölçmek için temel birim. Değişkenler, gerçek işlemlere dayalı olarak gerçek zamanlı olarak program tarafından hesaplanır. (Token Cost, Türkçe’de “Token Maliyeti” anlamına gelir.)

Aşağıdaki sabitler, parlamento organizasyonu ve geliştiriciler tarafından teklifler şeklinde birlikte yapılandırılabilir:

C: varsayılan maliyet ücreti, sayım 0 olduğunda, bu ücret yine de tahsil edilecektir

$k_1/k_2$ : sayımın birim fiyatı

changeSpan: sayım belirli bir miktarı aştığında, değişiklik doğruluğunu kontrol etmek için kullanılır

weight/weightBase: sayım belirli bir miktarı aştığında, fazla sayımın birim fiyatını ayarlamak için kullanılır

OKUMA kullanımı, işlemdeki veri tabanına yapılan okuma sayısı ile ölçülür ve sayım, okuma sayısıdır. Test ortamı tahminlerine göre bir işlemdeki ortalama sayım, 10 birimden azdır. Gerçek bir iş senaryosunda veri tabanı okuma gereksinimlerinin artacağı düşünüldüğünde, üç ücretlendirme aralığına bölünür. Sayım <10 olduğunda, bir kez okunan veri tabanının birim fiyatı 1/8 OKUMA Tokeni olur. Sayım <100 olduğunda, birim fiyat 1/4 OKUMA Tokenidir. Sayım >100 olduğunda fazla kısım; changeSpan değerinin 2 olarak ayarlandığı, weight değerinin 250 olarak ayarlandığı ve weightBase değerinin 40 olarak ayarlandığı sayım (count)/changeSpan değerinin karesi olarak hesaplanır. Segment aralığı, sözleşme yükseltilecek güncellenebilir.

- YAZMA (WRITE) Token:

$$token\ cost = C + k_1 \times count \quad count \leq 10$$

$$token\ cost = k_2 \times count \quad 10 < count \leq 100$$

$$token\ cost = k_2 \times count + \left( \frac{count}{changeSpan} \right)^2 \times \frac{weight}{weightBase} \quad count > 100$$

**Count (Sayım):** tüketilen YAZMA kaynaklarını ölçmek için temel birim. Değişkenler, gerçek işlemlere dayalı olarak gerçek zamanlı olarak program tarafından hesaplanır.

Aşağıdaki sabitler, parlamento organizasyonu ve geliştiriciler tarafından teklifler şeklinde birlikte yapılandırılabilir:

C: varsayılan maliyet ücreti, sayım 0 olduğunda, bu ücret yine de tahsil edilecektir

$k_1/k_2$ : sayımın birim fiyatı

changeSpan: sayım belirli bir miktarı aştığında, değişiklik doğruluğunu kontrol etmek için kullanılır

weight/weightBase: sayım belirli bir miktarı aştığında, fazla sayımın birim fiyatını ayarlamak için kullanılır

YAZMA kullanımı, işlemdeki veri tabanına yapılan yazma sayısı ile ölçülür ve sayım, yazma sayısıdır. Test ortamı tahminlerine göre bir işlemdeki ortalama sayım, 5 birimden azdır.

Gerçek bir iş senaryosunda veri tabanı yazma gereksinimlerinin artacağı düşünülduğünde, üç ücretlendirme aralığına bölünür. Sayım <10 olduğunda, veri tabanı yazma işleminin birim fiyatı 1/8 kaynak tokenidir. Sayım <100 olduğunda, birim fiyat 1/4 kaynak tokenidir. Sayım >100 olduğunda fazla kısım; changeSpan değerinin 2 olarak ayarlandığı, weight değerinin 250 olarak ayarlandığı ve weightBase değerinin 40 olarak ayarlandığı sayım (count)/changeSpan değerinin karesi olarak hesaplanır. Segment aralığı, sözleşme yükseltilerek güncellenebilir.

- DEPOLAMA (STORAGE) Tokeni:

$$token\ cost = C + k \times count \quad count \leq 1MB$$

$$token\ cost = k \times count + \left( \frac{count}{changeSpan} \right)^2 \times \frac{weight}{weightBase} \quad count > 1MB$$

**Count (Sayım):** DEPOLAMA kaynaklarının tüketimini ölçmek için temel birim. Değişkenler, gerçek işlemlere dayalı olarak gerçek zamanlı olarak program tarafından hesaplanır.

Aşağıdaki sabitler, parlamento organizasyonu ve geliştiriciler tarafından teklifler şeklinde birlikte yapılandırılabilir:

C: varsayılan maliyet ücreti, sayım 0 olduğunda, bu ücret yine de tahsil edilecektir

$k_1/k_2$ : sayımın birim fiyatı

changeSpan: sayım belirli bir miktarı aştığında, değişiklik doğruluğunu kontrol etmek için kullanılır

weight/weightBase: sayım belirli bir miktarı aştığında, fazla sayımın birim fiyatını ayarlamak için kullanılır

DEPOLAMA kullanımı, gönderilen işlemin parametresinin (bayt) boyutu ile ölçülür ve sayım, parametrenin boyutunu belirtmek için kullanılır. 1MB sabitini eşik olarak alınız. Sayım <1MB olduğunda 64 bayt, 1 DEPOLAMA kaynak tokeni tüketir. Sayım >1 MB olduğunda 1 MB'den büyük olan kısım; ChangeSpan'ın 100'e, weight değerinin 250'ye ve weightBase'in 300'e ayarlandığı sayım (count)/changeSpan değerinin karesi olarak hesaplanır. Segment aralığı, sözleşme yükseltilerek güncellenebilir.

- TRAFİK (TRAFFIC) Tokeni:

$$token\ cost = C + k \times count \quad count \leq 1MB$$

$$token\ cost = k \times count + \left( \frac{count}{changeSpan} \right)^2 \times \frac{weight}{weightBase} \quad count > 1MB$$

**Count (Sayım):** tüketilen TRAFİK kaynaklarını ölçmek için temel birim. Değişkenler, gerçek işlemlere dayalı olarak gerçek zamanlı olarak program tarafından hesaplanır.

Aşağıdaki sabitler, parlamento organizasyonu ve geliştiriciler tarafından teklifler şeklinde birlikte yapılandırılabilir:

C: varsayılan maliyet ücreti, sayım 0 olduğunda, bu ücret yine de tahsil edilecektir

$k_1/k_2$ : sayımın birim fiyatı

changeSpan: sayım belirli bir miktarı aştığında, değişiklik doğruluğunu kontrol etmek için kullanılır

weight/weightBase: sayım belirli bir miktarı aştığında, fazla sayımın birim fiyatını ayarlamak için kullanılır

TRAFİK kullanım miktarı, gönderilen işlemin parametre boyutu (bayt) ile ölçülür ve sayım, işlem boyutunu temsil eder. Test ortamı tahminlerine göre işlemdeki ortalama sayım, yaklaşık 300 bayttır. Sistem sözleşmesinin büyüklüğü göz önüne alındığında, bir eşik olarak 1 MB'lık sabit bir değer kullanılır. Sayım <1MB olduğunda, her 64 baytta bir TRAFİK tokeni kullanılır, yani  $k = 1/64$ . Sayım >1MB olduğunda alınan kaynak tokeni; changeSpanBase değerinin 100 olarak ayarlandığı, weight değerinin 250 olarak ayarlandığı ve weightBase değerinin 500 olarak ayarlandığı maliyetin karesine göre artar. Segment aralığı, sözleşme yükseltilecek güncellenebilir.

İstatistiklere göre işlemlerde ortalama OKUMA kullanımı 40 defa, ortalama YAZMA kullanımı 40 defa, ortalama DEPOLAMA kullanımı 300 bayt ve ortalama TRAFİK kullanımı 300 bayttır. Ortalama kullanıma sahip işlemler için 30 kaynak tokeni ücreti alınır. Toplam ağırlığı (weight) 1000 olarak alarak, her bir tokenin birim ağırlığı 250/40, 250/40, 250/300, 250/300'dür.

## 2.8.2 Kullanıcı Ücretlendirme Modeli

Kullanıcı Ücretlendirme Modeli, iki bölümden oluşan işlemin yürütülmesi sırasında kullanıcı tarafından yapılan maliyetleri ifade eder. Biri işlemin büyüklüğüne göre ücretlendirilir ve diğeri sözleşmede geliştirici tarafından belirlenir. Formüller aşağıdaki gibidir:

$$\begin{aligned} \text{token cost} &= ( C + k \times \text{count} ) \times T & \text{count} \leq 1M \\ \text{token cost} &= \left( k \times \text{count} + \left( \frac{\text{count}}{\text{changeSpan}} \right)^2 \times \frac{\text{weight}}{\text{weightBase}} \right) \times T & \text{count} > 1M \end{aligned}$$

**Count (Sayım):** gerçekleştirilen işlemlerin boyutunu ölçmek için temel birim. Değişkenler, program tarafından gerçek zamanlı olarak hesaplanır.

Kullanıcıların hayati çıkarlarını içerdiğinden, aşağıdaki sabitlerin parlamento organizasyonları (seçilen üretim düğümlerinden oluşan Aelf ağı için ana yönetim organizasyonlarının daha fazla ayrıntısı Bölüm 5'tedir), geliştiriciler ve referandum organizasyonları aracılığıyla yapılandırılması gerekir:

T: Alınan tokenin dönüşüm oranı (sabit)

C: varsayılan maliyet ücreti, sayım 0 olduğunda, bu ücret yine de tahsil edilecektir

k: sayımın birim fiyatı

changeSpan: sayım belirli bir miktarı aştığında, değişiklik doğruluğunu kontrol etmek için kullanılır

weight/weightBase: sayım belirli bir miktarı aştığında, fazla sayımın birim fiyatını ayarlamak için kullanılır

İşlem ücreti, gönderilen işlemin parametre boyutu ile ölçülür. Test ortamı tahminlerine göre işlemin parametre boyutu, ortalama olarak 300 bayttır. Boyut olarak 1 MB'lık bir eşik sabit olarak alınır. Boyut <1MB olduğunda birim fiyat, sabit bir değer olan 1/800 olarak ayarlanır. Boyut >1MB olduğunda, sayım değeri arttıkça birim fiyat da artar. ChangeSpan 100, weight 1 ve weightBase 1 olarak ayarlanır. Sabit T, alınan tokenlerin dönüşüm oranıdır ve 3/50 olarak ayarlanmıştır. Farklı dönüşüm oranlarına sahip farklı tokenlerle birden çok token yapılandırılabilir. Segment aralığı, bir yükseltme sözleşmesi şeklinde güncellenebilir.

Her bir işlemin ilgili kaynaklar tüketilmese bile ücrete tabi olacağı düşünüldüğünde, her bir ücret türü için  $C = 0.0001$  maliyet ücreti alınır. changeSpan, yatay ölçeklendirme ayarları için kullanılır.

## 3. Teşvik Modeli - Düğümler ve Seçmenler

### 3.1 Ödül Havuzu

Aelf, ödülleri aşağıdaki bileşenlerden oluşan bir ödül havuzu aracılığıyla dağıtır:

- Blok Ödülü:

Aelf vakfı, üretilen blok sayısına göre üretim düğümlerine verilen bir kısmı ile toplam ELF tokenlerinin %12'sini topluluk yönetim ödülleri için Whitepaper'a uygun olarak piyasaya çıkardı. Ödül miktarı, her 4 yılda bir %50 azaltılır. Aelf Blockchain, her 4 saniyede 8 blok üretir ve her blok, başlangıçta 0.125 ELF üretir.

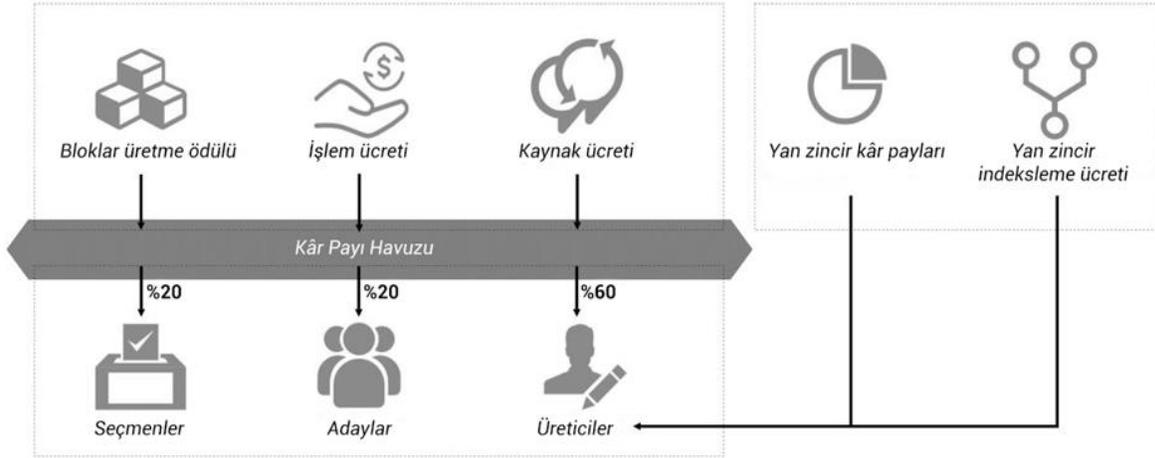
Zaman	Ödül
1. yıldan 4.yıla	Üretilen her bir blok için 0.125 ELF
5. yıldan 8. yıla	Üretilen her bir blok için 0.0625 ELF
9. yıldan 12. yıla	Üretilen her bir blok için 0.03125 ELF

- İşlem Ücreti:

İşlem ücretlerinin %90'ı ödül havuzuna girer.

- Kaynak Tokenleri İşlem Ücreti:

Kaynak tokenlerinin ticareti, işlem ücretlerini oluşturacaktır ve işlem ücretlerinin %50'si ödül havuzuna girecektir



Şekil 1.3 Ödül Havuzu Yapısı

Dağıtım yöntemi ve kâr payı/temettü havuzunun oranı:

Türler	Kâr Payı/Temettü Havuzunun Yüzdesi
Üretim Düzümü Temel Geliri	40%
Üretim Düzümü Oy Ağırlıklı Geliri	10%
Üretim Düzümü Yeniden Seçim Geliri	10%
Aday Düzümü Geliri	20%
Seçmen Geliri	20%

## 3.2 Üretim Düzümleri Teşvikleri

Üretim düğümlerinin sorumlulukları:

1. Blokları diğer düğümlere yayınlarak işlemlerin bloklara toplanması, doğrulanması ve paketlenmesi;
2. Doğrulamadan sonra tüm Aelf ekosisteminin kararlılığını sağlamak için blok, yerel blok zincirine eklenecektir.

Teşvikler:

1. Önceki oturumda üretilen blok sayısı, üretim düğümleri için dağıtım kar ağırlığı olarak kullanılır, ödül havuzunun "Üretim Düğümü Temel Geliri" dağıtır. (Ödül havuzunun %40'ı);
  2. "Üretim Düğümü Oy Ağırlıklı Geliri", her bir seçim turunda üretim düğümünün aldığı oyların ağırlığına göre tahsis edilir (ödül havuzunun %10'u);
  3. "Üretim Düğümü Yeniden Seçim Geliri", bir üretim düğümünün seçildiği ardışık terimlerin sayısına göre tahsis edilir. Tüm üretim düğümleri için başlangıç dağıtım indeksi 1'dir ve her bir ek terim, dizini 1 artırır. Yeniden seçim indeksinin üst sınırı 10'dur (ödül havuzunun %10'u);
  4. Geliştiriciler için bir yan zincir ağı oluşturmanın faydaları:
    - a. Yan zincir geliştiricileri için endeksleme blokları için indeks ücreti geliri;
    - b. Yan zincir geliştiricilerinden kaynak ücretlerinin alınması;
    - c. "Yan zincir destek planının getirdiği "temettü geliri".
- Not: Madde 4 "Teşvik Modeli - Yan Zincir Geliştiricileri" bölümünde daha ayrıntılı olarak açıklanmaktadır.

### 3.3 Aday Düğüm Teşvikleri

Aday düğümlerin sorumlulukları: Blok doğrulama, üretim düğümünün değiştirilmesi ve tüm Aelf ekosistem kararlılığının sağlanması.

Teşvikler: Elde edilen oy sayısına (üretim düğümlerinin sayısı \* 5) göre, "aday düğüm geliri" kısmı eşit olarak dağıtılır (ödül havuzunun %20'si).

### 3.4 Seçmen Teşvikleri

Seçmenlerin sorumlulukları: topluluğun iradesini temsil etmek için üretim düğümlerine oy verilmesi

Teşvikler: Teşvik oranı, oy kullanan kullanıcıların sayısının ağırlığına ve belirtilen kilit süresine göre tahsis edilir. Bir düğüme oy vermek için kullanıcıların oylarını kilitleyecekleri süreyi seçmeleri gerekir. Kullanıcılar 3, 6, 12, 24 veya maksimum 36 aylık kilit süresini seçebilir. Dağıtım, daha büyük ödül için daha uzun kilit süreleri prensibine dayanacaktır.

Kilit süresinin oy sayısına oran ağırlığı 2:1'dir.

Her bir oyun ağırlığı, kilitlenen gün sayısına göre bileşik faiz ile ilgili olarak hesaplanır.

Günlük faiz oranı, farklı kitleme süreleri için farklıdır:

- Bir yıllık günlük faiz oranı: %0,1
- İki yıllık günlük faiz oranı: %0,15
- Üç yıllık günlük faiz oranı: %0,2

Weight (Ağırlık), aşağıdaki formüle göre hesaplanır:

$$total\ weight = amount \times \left( (1 + interest)^t + \frac{1}{2} \right)$$

*amount (miktar)*, kullanıcı oylarının değişken sayısıdır. *t* değişkeni, kullanıcılar tarafından belirlenen iki değişken olan kilitleme süresine (gün cinsinden) karşılık gelen oydur.

Oy kullanan tüm kullanıcılar, ödülleri yukarıdaki formüle göre hesaplayabilir. Oy temettü gelir havuzunun kendi kısımlarını alan atanmış bir kullanıcı olacaklardır. (Temettü havuzunun %20'si)

Parlamento organizasyonları aşağıdaki sabitleri teklif şeklinde değiştirebilirler: faiz (interest): her oy; bir yıllık, iki yıllık veya üç yıllık kilitleme dönemi günlük faiz oranlarına göre belirlenen günlük oranlarda kilitlenir.

## 3.5 Düğüm Sayısı

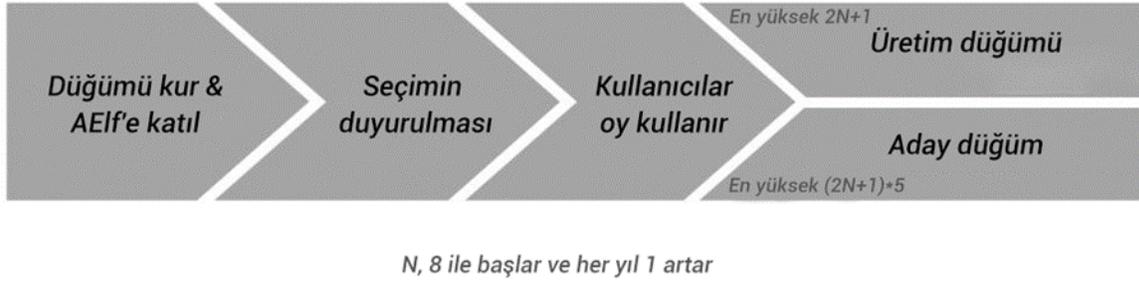
Aelf Whitepaper'a göre ilk aşamada, aday düğümlerden 17 üretim düğümü seçilecektir. Zincir sağlam bir şekilde çalıştıktan sonra, her yıl 2 yeni üretim düğümü eklenecektir. Maksimum üretim düğümü sayısı, ekosistem gelişiminin ihtiyaçlarını karşılamak için ana ağa oy verdikten sonra topluluk tarafından belirlenecektir.

## 3.6 Üretim Düğümü Seçimleri

### 3.6.1 Seçim Gereksinimleri

- Bir düğüm adayı tarafından yatırılması gereken ELF tokenlerinin sayısı:  $\geq 100,000$  ELF.
- Yapılandırma önerileri: 8 çekirdekli işlemci, 16 GB bellek, 500 GB sabit disk, blok ve işlem senkronizasyonunu sağlamak için yeterli bant genişliği.
- Düğüm coğrafi konumu: bütün dünyada  
Not: Bir üretim düğümü başarılı bir şekilde seçildikten sonra yatırmış olduğu tokenler kilitlenecektir. Kilitli tokenler, işlem ihlali olmadığı varsayılarak, üretim düğümünü çalıştırmaktan vazgeçtikleri zaman düğüm sahibine iade edilecektir.

### 3.6.2 Seçim Süreci



Şekil 3.7 Düğüm Seçim Prosedürü

1. Bir üretim düğümü olmak isteyen ekipler veya bireyler, başarılı bir şekilde bir düğüm oluşturduktan ve onu Aelf Blok Kaşifine (Block Explorer) bağladıktan sonra seçime katılabilir
2. Topluluğa aday düğümlerin bir listesi duyurulacak ve üretim düğümlerine oylarla en iyi 17 düğüm seçilecek.
3. Üretim düğümleri, bir rotasyon sistemi içerisinde yer alır ve her hafta yeni bir üretim düğümü seçimleri yapılacaktır. Düğümler, alınan oy sayısına göre en yüksekten en düşüğe doğru listelenir.

### 3.7 Oylama Kuralları

Seçmenler, bir ELF tokeninin bir oya eşit olduğu oranla ELF kullanarak düğüm seçimlerine katılabilirler. Bir bireyin kullanabileceği oy sayısı veya oy kullanabileceği düğüm sayısı konusunda herhangi bir kısıtlama yoktur.

1. Seçmenler için varsayılan kilit süresi 3 aydır ve kullanıcılar kilitli süre boyunca ödülleri kaybetmeden tokenleri çekemezler.
2. Kullanıcılar; 3, 6, 12, 24 veya 36 aylık seçenekler arasından kilitli süreyi özgürce seçebilir;
3. Kilitli süre boyunca kullanıcılar, hangi düğümün oylarını alacağını serbestçe değiştirebilir.
4. Oylama ödülleri, kullanılan oy sayısına ve seçilen kilitlenme süresine bağlı olarak değişir. Kilitlenme süresi ne kadar uzun olursa, kâr payı/temettü ağırlık ödülü o kadar yüksek olur;
5. Kilitli süre sona erdikten sonra, kilitli tokenler kullanılabilir ve artık kâr payı kazanmaz.

### 3.8 Üretim Düğümü Eliminasyon Mekanizması

#### 3.8.1 Seçim Dönemi

Her hafta üretim düğümü seçimlerinin yeni bir turu yapılacak ve teminat/depozito hâlâ kilitli olduğu sürece mevcut üretim düğümü otomatik olarak bir sonraki seçim turuna girecektir.

### 3.8.2 Eliminasyon Kuralı

Bir üretim düğümü 72 saat içinde herhangi bir blok üretmezse, üretim düğümünün niteliği otomatik olarak iptal edilecek ve yatırılan ELF imha edilecektir.

Bir üretim düğümü kötü amaçlı bir işlemi paketlerse, diğer düğümler aldıkları bloğu doğrulayamaz ve kötü amaçlı işlemi paketleyen üretim düğümü ile ağ bağlantısını keser. Bu üretim düğümü ayrıca kara listeye de eklenecektir.

## 3.9 Token Sahibi Sözleşmesinde Ödüller

Aelf'de geliştiriciler, sözleşmeleri için token çıkarabilir ve bu tokeni elinde tutan kullanıcılar ilgili sözleşmenin ödüllerinden yararlanabilir. Spesifik adımlar aşağıdaki gibidir:

1. Bir token sahibi ödül planı oluşturmak için geliştirici sözleşmesindeki Token Sahibi sözleşmesinin kullanılması;
2. Geliştirici sözleşmesinde vaat edilen ödül, Token Sahibi sözleşmesi tarafından sağlanan yöntemle ödül planına girilir;
3. İlgili tokeni elinde tutan kullanıcı, kâr payı ağırlığını ödül planından kilitleme yoluyla alır;
4. Geliştiriciler ödül topladığında, ödüllerin bir kısmını otomatik olarak serbest bırakırlar ve kilitlemeye katılan kullanıcılar ödül alabilir.

## 4. Teşvik Modeli - Yan Zincir Geliştiricileri

Aelf'in 'ana zincir + çoklu yan zincirler' modeli, her bir yan zincirin "bir zincir, bir senaryo" tasarımını benimsemesi ile genel ekosistemin çekirdeğidir. Bu tasarımın amacı, kaynakların etkin bir şekilde ayrılmasını sağlamak ve platforma ölçeklendirme yetenekleri sağlamaktır. Bu bölümde, yan zincir geliştiricileri için teşvik modeli açıklanmaktadır.

### 4.1 Yan Zincir Oluşturma

Geliştiriciler, işletim verimliliğini sağlamak için bağımsız bilişim kaynakları gerektiren bir uygulama oluşturduğunda; bunun için ayrı bir yan zincir oluşturmayı düşünmek gerekir.

Kullanıcıların yan zincirin performansını artırması/azaltması gerektiğinde, sisteme başvurabilirler. Bu süreç, gereksiz kaynak israfını önleyerek bir bulut servis sağlayıcısının elevatör sunucusu kadar uygundur/kullanışlıdır.



Şekil 4.1 Yan Zincir Oluşturma Prosedürü

1. Bir hesap oluřturunuz
2. Hesaba belirli miktarda ELF yatırınız; (Spesifik miktar üretim düğümleri tarafından kararlařtırılır)
3. Bir yan zincir uygulaması oluřturulmasını bařlatmak için sözleşmeyi çağırınız
4. Üretim düğümü oylaması (üretim düğümü oylarının 2/3'ünden fazlasını almanız gerekir)
5. Üretim düğümleri bir yan zincir oluřturma talebini kabul ederse, yan zincir düğümünün oluřturulması 3 gün içinde tamamlanacaktır.

Yan zincir bařarıyla oluřturulduktan sonra yatırılan ELF, indeks ücretini düşürmek için kullanılacaktır (indeks ücreti tutarı, üretim düğümleri tarafından ortaklařa belirlenir). Geliřtiricilerin, yan zincirin normal olarak indekslenmesi için hesap bakiyesinin yeterli olduğundan emin olması gerekir.

## 4.2 Yan Zincir Kapatma Mekanizması

Bağımsız bir yan zincir geliřtiricisinin herhangi bir nedenle bir yan zinciri kapatması gerektiğinde, üretim düğümleri tarafından incelenmesi için bařvurmaları gerekir. İnceleme üzerine üretim düğümleri, yan zincir kapanmasını tamamlama yeteneğine sahiptir.

Yan zincirdeki indeks ücreti hesabının bakiyesi yetersiz olduğunda, ana zincir yan zincirdeki indekslemeyi otomatik olarak durduracaktır. Bu durumda üretim düğümleri, belirli yan zincirin hizmetlerini durdurma hakkına sahiptir.

Risk uyarısı:

- İndeks işlevinin durdurulması, çapraz zincir transferlerinin ve çapraz zincir doğrulamasının yapılamayacağı ve bu yan zincirdeki kullanıcıların tokenlerinin aktarılamayacağı anlamına gelir.

## 4.3 Yan Zincir Geliřtirici Ücretlendirme Modeli

### 4.3.1 Özel Yan Zincir Ücretlendirme Modeli

Özel bir yan zincirin ücretlendirme modeli ařağıdaki gibidir:

1. Zaman başına ödeme modeli, gerçek blok üretim süresine göre özel kaynak kullanım ücretini (CPU kaynağı / RAM kaynağı / DISK kaynağı / NET kaynağı) öder. Bu, bir uygulamayı bir bulut sunucusuna dağıtması gereken bir geliřtirici gibidir. Bulut sunucusunun yapılandırmasına göre ödeme yapılır (CPU / RAM / DISK ve sabit bant genişliğı);
2. Çapraz zincir indeks ücret modelinde çapraz zincir transfer/doğrulama yeteneklerini saėlamak için bir yan zincir oluřtururken, indeks ücretini ödemek için belirli bir miktar ELF'in ipotek edilmesi gerekir. İndeks ücreti, blok sayısına göre tahsil edilir ve birim fiyat, parlamento organizasyonu ile geliřtirici arasındaki müzakere yoluyla belirlenir. Ana zincir üretim düğümü; bir yan zincir düğümü indekslemesi gerçekleřtirerek yan zincirin çapraz zincir aktarımı ve doğrulama yeteneklerine sahip olmasını saėlar, yan zincirin işlevlerini genişletir ve yan zincir ağıının güvenliğini saėlar. Ana zincir üretim düğümü, bunun için indeks ücretini alır.

Özel bir yan zincir, geliştiricilerin işlem ücreti modelini seçmelerine ve işlem ücret fiyatını ve işlem ücreti alma adresini belirlemelerine olanak tanır.

### 4.3.2 Yan Zincir Destek Planı Ücretlendirme Modeli

İlk zaman başına ödeme modelinden feragat etmek için yan zincir geliştirici destek planı için öneri

1. Aşağıdaki ücret modelini kullanarak özel yan zincir geliştiricilerinin yeniliklerini desteklemek için 4 yan zincir kurulması:
  - a. Geliştiriciler Paylaşım Modeli: geliştiriciler, belirli bir gelir yüzdesini üretim düğümleriyle paylaşmaya söz verir;
  - b. Çapraz zincir indeks ücret modeli (özel yan zincir ücreti modeliyle aynı)
2. "Destek Planı" için başvuran geliştiricilerin, üretim düğümleri tarafından incelenecek bir proje planı ve paylaşım planı sunmaları gerekmektedir. İncelemeyi geçtikten sonra üç ay boyunca "Yan Zincir Destek Planı'nı" kullanabilirler;
3. Üç ay sonra üretim düğümü, desteği sürdürmek veya durdurmak için oy kullanacaktır;
4. Destek durdurulduktan sonra üretim düğümü, bu yan zincirin hizmetlerini durdurma hakkına sahiptir.

### 4.3.3 Paylaşım Yan Zinciri Ücretlendirme Modeli

Yan Zincir Paylaşım teklifi: Herhangi bir geliştiricinin üzerinde bir sözleşme dağıtılabileceği paylaşılmış bir yan zincir kurunuz. Paylaşılmış yan zincir, geliştirici ödemeli işlem ücreti modelini veya geliştirici gelir paylaşım modelini + kullanıcı ödemeli işlem ücreti modelini kullanmayı seçebilir.

Geliştirici ödemeli işlem ücreti modeli: Geliştiricinin sözleşmelerinin sözleşme çalışırken sürekli kaynak tüketmesi gerekir. Geliştiriciler, tüketilen kaynaklara göre ücret ödemek zorundadır. Bu, geliştiricileri programı optimize etmeye ve kaynak tüketimini en aza indirmeye motive edebilir. Tutulan kaynak tokenleri bittiğinde, bu sözleşmenin işlem yürütmesi başarısız olur ve geliştiricinin sözleşmeyi sürekli çalışmasını sağlaması için yeterli miktarda kaynak tokenini sağlaması gerekir. Geliştirici işlem ücreti, paylaşılmış yan zincir konsensüs sözleşmesine girilerek üretim düğümlerine eşit olarak dağıtılacaktır.

Geliştirici gelir paylaşım modeli + kullanıcı ödemeli işlem ücreti modeli: Yan zincirin güvenliğini paylaşmak için geliştirici, gelir paylaşım modelini seçerse kullanıcı ödemeli işlem ücreti modeli de seçilmelidir. Geliştirici tarafından vaat edilen gelir payı, önceden belirlenmiş bir oranda üretim düğümlerine dağıtılır. Kullanıcı işlem ücretleri, paylaşılmış yan zincirin oluşturucusu tarafından belirtilen adresi girer.

Paylaşılmış yan zincirin oluşturucusu, çapraz zincir transfer ve doğrulama işlevlerini sağlamak için bir çapraz zincir indeks ücreti ödemek zorundadır.

## 4.4 Üretim Döğümleri Teşvikleri

### 4.4.1 Yan Zincirlerde Sorumluluklar

- Geliştiricilerin uygulamalarının kararlı bir şekilde çalışmasına olanak tanıyan yan zincir döğümleri oluşturma ve bakımını yapma
- Yan zincirin güvenliğini sağlama

### 4.4.2 Yan Zincir Ekosistemlerinde Ödüller

- Blok indeks ücretleri, geliştiriciler tarafından blok birimleri halinde ödenir. İndeks ücretinin birim fiyatı, yan zincir tarafından seçilen kaynak türüne göre değişir ve üretim döğümleri tarafından kabul edilir
- Uygulamanın normal çalışmasını sağlamak için geliştiriciler tarafından ödenen kaynak ücretleri. (Yan zincir destek planına katılıp katılmadıklarına bakılmaksızın, işlem ücretleri olacaktır)
- Geliştirici yan zincir destek planına katılırsa üretim döğümleri, geliştirici tarafından vaat edilen “gelir kâr payının” bir kısmını alır
- İndeks ücretleri, kaynak ücretleri ve kâr payları; bir sonraki seçimin bitiminden önce tüm üretim döğümlerine eşit olarak dağıtılacaktır.

## 5. Yönetişim

Yönetişim, tüm toplum için çok önemlidir ve toplumun sürdürülebilir ve sağlıklı bir şekilde işlemlerini sağlamak için temel oluşturur. Aelf, üç yönetim modeli sunar. Bu yönetim araçları aracılığıyla kullanıcılar, aday döğümler ve üretim döğümleri çeşitli işlerin yönetimine tam olarak katılabilir ve tüm ekosistemi daha adil hale getirmek için kendi rollerini aktif olarak oynayabilirler.

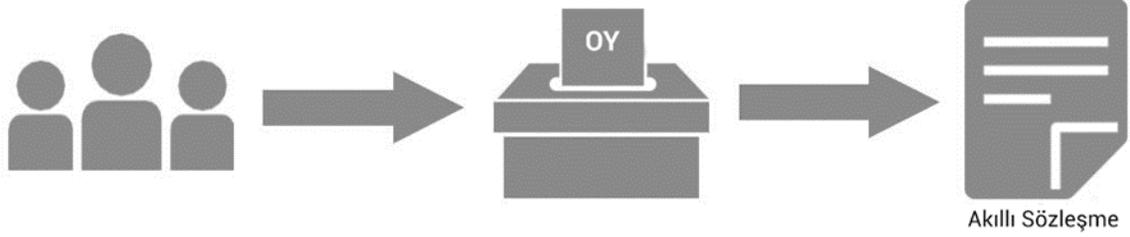
Bu üç yönetim modeli, Aelf'in ana zincirinin ve yan zincirlerinin yönetiminde önemli bir rol oynamıştır ve ayrıca yan zincir geliştiricilerinin kendi senaryolarını birleştirerek kendi yönetim sistemlerini kurmaları için güçlü araçlardır.

Yönetişim, yönetim otoritesinin taşıyıcısı olan organizasyon biçimi ile sağlanır. Yönetişim hakları, organizasyonun rızasıyla diğer organizasyonlara devredilebilir ve yeni organizasyon yönetim haklarını kullanmak ve uygulamak için eski organizasyonu devralacaktır.

## 5.1 Yönetişim modelleri

### 5.1.1 Parlamento Yönetişim Modeli

Kullanıcılar, çoğu kullanıcı adına işlemel yönetişim haklarını kullanan üretim düğümüne oy verir. Aelf ağında üretim düğümleri, önemli konuları yönetmek için Parlamento modelini kullanır. İlk durumda üretim düğümleri, Parlamento üyeleridir ve sadece üretim düğümlerinin üçte ikisi belirli bir karar lehine oy verdiğinde o karar yürütülecektir.



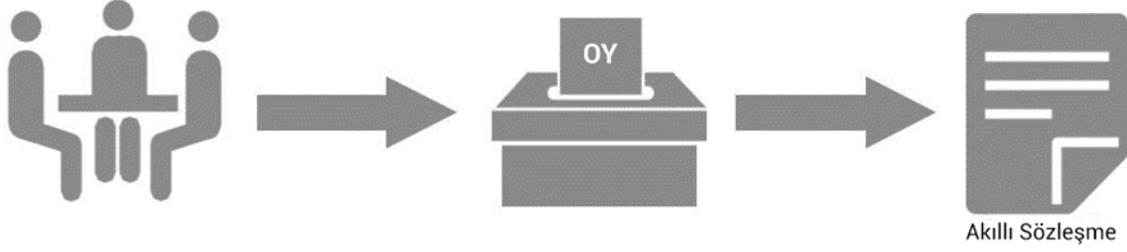
Şekil: 5.1.1 Parlamento Yönetişim Modeli

Parlamento üyeleri, yeni bir Parlamenter Kuruluş oluşturmak ve kullanmak için aşağıdaki adımları izleyebilir:

1. Bir parlamenter organizasyon oluşturulması ve kararların yürütülmesini tetikleyen koşulların ayarlanması (üyeler, üretim düğümleridir);  
Tetikleyici koşullara örnekler: üretim düğümlerinin 2/3'ten fazlası evet oyladı, 1/5'ten az düğüm karşı oy kullandı ve 1/5'ten az düğüm bulunmadı ve katılımcı sayısı 3/4'ten az değildi.
2. Parlamento organizasyonu üyeleri, görüşlerini ifade ederler: lehine, aleyhine veya çekimser;
3. Belirlenen koşullara ulaşıldığında, kararın yürütülmesi tetiklenir.

### 5.1.2 Birlik/Ortaklık Yönetişim Modeli

Büyük bir organizasyon olan tüm topluluğa ek olarak, Aelf platformunda birçok küçük organizasyon olacaktır. Bu organizasyonlar, belirli hedeflere ulaşmak için kurulmuşlardır ve aynı zamanda organizasyon içinde işlemleri iş birliği içinde yürütmek için DAO (dağıtılmış Otonom organizasyonlar) vb. etkili yönetim araçlarına ihtiyaç duyarlar. Aelf, özellikle ortaklık içindeki yönetişimi ele alan bir ortaklık yönetişim modeli sağlar.



Şekil: 5.1.2 Birlik/Ortaklık Yönetişim Modeli

Ortaklık Yönetişimi aşağıdaki gibi kullanılır:

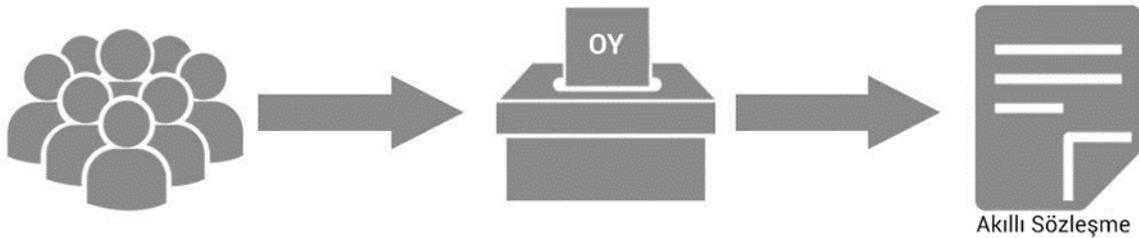
1. Ortaklık organizasyonları oluşturulması, ortaklık organizasyonlarının üyelerinin eklenmesi ve işlemin yürütülmesini tetikleyen koşulların ayarlanması;  
Tetikleyici koşullara örnekler: Ortaklıkta 5 üye vardır, sadece 3'ten fazla üye lehte oy verir, 2'den az üye aleyhte oy verir, 2'den az üye oy kullanamaz ve en az 4 üye işlemin yürütülmesine katılır.
2. Ortaklık üyeleri, görüşlerini ifade ederler: evet, hayır veya çekimser;
3. Belirlenen koşullara ulaşıldığında, işlemin yürütülmesi tetiklenir.

### 5.1.3 Referandum Yönetişim Modeli

Üretim düğümleri veya ortaklıklar, tüm kararları belirleyemez. Bazı son derece önemli kararlar, özellikle de kullanıcı hakları ve çıkarlarını içeren kararlar, tüm kullanıcıları kapsmalıdır ve kullanıcının yönetim için oy kullanmasına tam kontrol verilmelidir. Referandum yönetim modeli bunun için oluşturuldu.

Aelf'de iki referandum yönetim modeli vardır. Birincisi, belirlenen şartların yerine getirilmesinden sonra kararın uygulanmasıdır (evet/ hayır/çekimser); ikincisi ise seçenekleri ayarlamaktır ve kullanıcılar isteklerine göre seçenekleri seçebilirler.

İlk referandum modeli aşağıda gösterilmiştir:

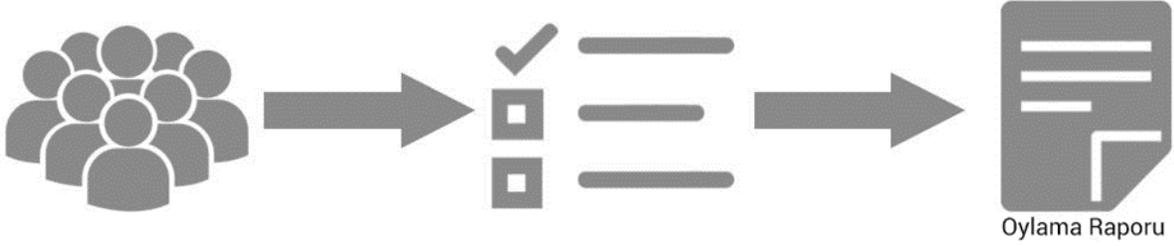


Şekil: 5.1.3-1 Referandum Yönetişim Modeli 1

Referandum yönetim modelini kullanmak için aşağıdaki adımları izleyiniz:

1. Bir referandum organizasyonun oluşturulması, referandum token türünün ve tetikleyici koşulların (ipotek tokenlerinin sayısı) ayarlanması;  
Tetikleyici koşullara örnekler: Toplam oy sayısı 1.000.000'dan az değildir, lehte oylar 500.000'den az değildir, aleyhte oylar 100.000'den azdır ve çekimserlerin sayısı 100.000'den azdır.
2. Kamuoyuna tokenlerini kullanarak oylamaya katılmaları ve görüşlerini belirtmeleri için duyuru: evet, hayır veya çekimser;
3. Belirlenen koşullar karşılandığında, belirli bir kararın uygulanması tetiklenir.

İkinci referandum modeli aşağıda gösterilmiştir:



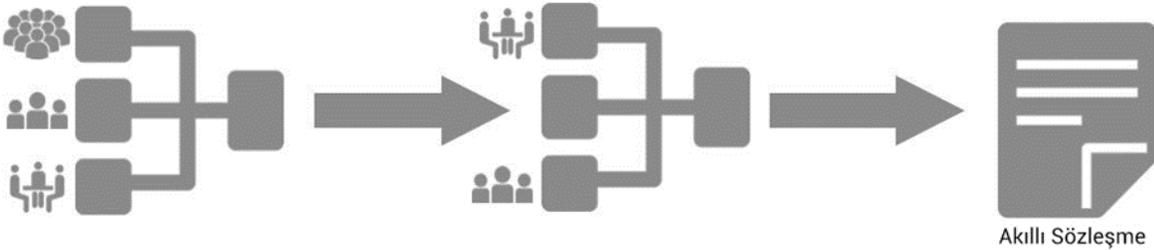
Şekil: 5.1.3-2 Referandum Yönetişim Modeli 2

Referandum yönetim modelini kullanmak için aşağıdaki adımları izleyiniz:

1. Oylama öğelerinin oluşturulması ve referandum seçeneklerinin ayarlanması: oylama için kullanılan token türü, kilitli olup olmadığı, oylamanın başlangıç - bitiş zamanı, vb.; Oylama örneği: Oylama için token türünü ayarlanması, doğruya (true) kilitlenmesi, oylama başlangıç zamanı şimdi ve bitiş zamanı bir hafta sonra. Süre dolduğunda oylama otomatik olarak sona erer;
2. Kamuoyuna referandum seçeneklerine oy vermeleri için duyuru;
3. Oylama süresi dolduktan sonra oylama tamamlanır.

#### 5.1.4 Özelleştirilmiş Yönetim Modeli

Yukarıda belirtilen üç yönetim modeline ek olarak daha karmaşık yönetim senaryolarını karşılamak için Aelf, aynı veya farklı yönetim modellerinin çoğunu seri veya paralel olarak karmaşık özelleştirilmiş yönetim modelleri oluşturmak üzere birbirine bağlayarak yönetimi kademelendirme yeteneği sağlar. Kararlar, ancak her organizasyon kabul ederse uygulanabilir.



Şekil: 5.1.4 Özel Yönetişim Modeli

Karmaşık bir yönetim modeli kullanmak için şu adımları izleyiniz:

1. Gerçek yönetime dayanarak organizasyonların (parlamento organizasyonları, ortaklık organizasyonları, referandum organizasyonları) oluşturulması;
2. Mevcut duruma göre, çeşitli organizasyonlar paralel veya seri olarak bağlanır;
3. İlgili organizasyon üyelerinin yönetime katılmalarında görüşlerini ifade etmeleri için duyuru;
4. Belirlenen eşiğe ulaşıldığında, nihai kararın yürütülmesi tetiklenir.

## 5.2 Mevcut Ana Zincir Yönetimi

Üç temel yönetim modeli, ana zincirin yönetimini kolaylaştırır. Bu bölümde, bu üç temel yönetim modelinin ana zincir işlemlerini yönetmek için nasıl kullanılacağı açıklanmaktadır.

### 5.2.1 Parlamento Yönetim Modelini Kullanma

Parlamento yönetim modeli için tetikleyici koşullara örnekler: Parlamento örgütlerinde yalnızca lehte oylar  $2/3$ 'ü aştığında, aleyhte oylar  $1/5$ 'ten az olduğunda, çekimser oylar  $1/5$ 'ten az olduğunda ve oy sayısı  $3/4$ 'ten az değilse bir kararın yürütülmesi tetiklenir.

Parlamento modelinin yönettiği işler:

1. Ana zincir sözleşme dağıtımı/yükseltmesi  
Otomatik sözleşme denetimleri olmasına rağmen, güvenlik için, ana zincir dağıtım ve yükseltme sözleşmelerinin başlatılmasının ilk aşamasında parlamento tarafından onaylandıktan sonra dağıtılması/yükseltilmesi gerekmektedir. (Bu aynı zamanda ana zincir DApp yönetim sürecidir)
2. Üçüncü parti bir token ve ELF konektörü oluşturma  
Aelf'de ELF stake yaparak ELF ile bir konektör oluşturabilir ve ELF ile değişimi tamamlayabilirsiniz. Bu kural, Bancor modeline dayanmaktadır. Bağlantıları kuran ve etkinleştiren tokenler, işlem için Bancor modelini takip edecektir. Konektörlerin oluşturulmasını ve etkinleştirilmesini talep eden operasyonlar, parlamenter organizasyon yönetimini kullanır.
3. Kullanıcı oyları için ödül oranının ayarlanması  
Oy kullanan kullanıcılar, oyların kilitlenme süresine bağlı olarak ağdan ödüller alacaktır. Daha uzun süreler daha yüksek ödüllerle sonuçlanır. Oranın ayarlanması durumunda, Parlamento'nun günlük ödül oranını bir, iki ve üç yıl için ayarlamayı kabul etmesi gerekir. Spesifik ayarlanabilir parametre aralığı için lütfen 3.4'e bakınız.
4. Sistem sözleşmesi yöntem ücretinin ayarlanması  
Sistem sözleşmesi işlem ücreti, iki bölümden oluşur: biri işlemin büyüklüğüne göre ücretlendirilirken, diğeri sözleşme yöntem ücreti olarak parlamento tarafından belirlenir.
5. İşlem ücretleri için ödeme olarak kullanılabilen mevcut tokenlerin listesi

İşlem ücretleri, sadece ELF ile değil aynı zamanda seçilen diğer tokenler (stabil koinler gibi) kullanılarak da ödenebilir. Kullanılabilecek tokenlerin listesi, yapılandırılabilir ve parlamento tarafından birlikte yönetilebilir.

### 5.2.2 Birlik/Ortaklık Yönetişim Modelini Kullanma

Ortaklık, özerk/otonom bir organizasyondur ve ortaklık içindeki işler ortaklık yönetişimi yoluyla iş birliği içinde belirlenebilir, örneğin:

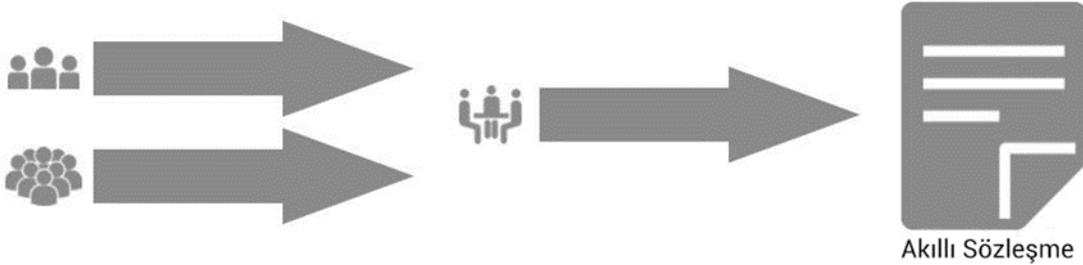
1. Ortaklık üyeleri hesap varlıklarını birlikte yönetir  
Ortaklık yönetişim modeli tetikleyici koşulları örneği: Ortaklıkta 5 üye vardır, sadece 3'ten fazla üye lehte oy verir, 2'den az üye aleyhte oy verir, 2'den az üye çekimser kalır

### 5.2.3 Referandum Yönetişim Modelini Kullanma

Referandum yönetişim modeli tarafından yönetilen işler şunlardır:

1. Üretim düğümleri ve aday düğümlerin seçimi  
Kullanıcılar, isteklerini temsil edebilecek üretim düğümleri ve aday düğümler için oy kullanırlar ve aynı zamanda oylama gelirini alırlar. Üretim düğümleri ve aday düğümlerin seçimi, bir referandum organizasyonu aracılığıyla gerçekleştirilir. Bu referandum, kullanıcıların seçebileceği seçenekleri (seçime girmek için oylamaya girme seçeneği) listeler.

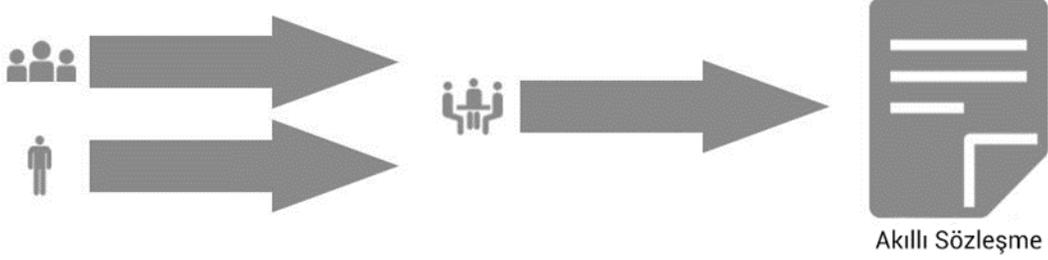
### 5.2.4 Özel Bir Yönetişim Modeli Kullanma



Şekil: 5.2.4-1 Özel Yönetişim Modeli (parlamento organizasyonları, referandum organizasyonları, ortaklık organizasyonları)

Yukarıdaki şekilde özel yönetişim modelini göstermek için bir örnek alınız: parlamento organizasyonlarını, referandum organizasyonlarını ve dernek organizasyonlarını içerir. Parlamento organizasyonları ve referandum organizasyonları parlamento organizasyonlarının üyeleridir. Ortaklık organizasyonlarında, değişiklikleri tetiklemek için yalnızca iki üye onay için oy kullanabilir. Parametrelerin işleri; parlamenter organizasyonda sadece lehte oylar 2/3'ten fazla olduğunda, oylar 1/5'ten az olduğunda, çekimserler 1/5'ten az olduğunda ve katılım oranı 3/4'ten az olmadığında. Evet, referandum organizasyonu toplam 1.000.000'dan az oya sahip değildir, lehine

oy 500.000'den az değildir, aleyhte oy 100.000'den fazla değildir ve ortaklık organizasyonunda 100.000'den fazla çekimser oy yoktur.



Şekil: 5.2.4-2 Özel Yönetişim Modeli (parlamento organizasyonları, geliştiriciler, ortaklıklar)

Yukarıdaki şekilde özel yönetim modelini göstermek için bir örnek alınız: Parlamento organizasyonlarından, geliştiricilerden ve ortaklık organizasyonlarından oluşur. Şekil 5.2.4-1'den farkı, ortaklık katılımcılarının konseyi ve geliştiriciyi içermesidir. İşler, ancak her iki üyenin de lehte oyu ile uygulanabilir.

Bir özel yönetim modeli kullanan işlemler şunlardır:

1. Kullanıcı oylama geliri faiz oranının ayarlanması  
Kullanıcılar, oy vererek gelir elde edebilirler. Oylama kilit süresi ne kadar uzun olursa, o kadar fazla gelir elde edilir. Özel ayarlanabilir parametre aralığı için lütfen 3.4'e bakınız.
2. Kullanıcı ücretlendirme modelinde önemli parametrelerin ayarlanması  
Kullanıcıların ücretlendirilmesi; kullanıcıların, geliştiricilerin ve üretim düğümlerinin çıkarlarını içerir ve kullanıcılar, geliştiriciler ve üretim düğümleri tarafından ortaklaşa belirlenir. Bu, üç yönetim modeline dayanan özel yönetim modelleri gerektirir. Belirli ayarlanabilir parametre aralıkları için lütfen Şekil 2.5.2'ye bakınız.
3. Geliştirici ücreti modelinde önemli parametreleri ayarlanması  
İkinci bölüm, işlem ücreti modelini ifade eder. İşlem ücreti modeli, bir geliştirici ödeme modeline ve bir kullanıcı ödeme modeline ayrılmıştır. Bu modeller çıkarsama yapılmış formüllerden oluşur. Bunlar arasında, ücretleri ayarlamak için parametreler vardır. Parlamenter organizasyon ve geliştirici, kooperatif bir organizasyon oluşturur ve birlikte karar verir. Özel ayarlanabilir parametre aralığı için lütfen 2.5.1'e bakınız.

### 5.3 Yan Zincir Yönetişimi

Üç yönetim modeli, ayrıca yan zincir yönetişimini kolaylaştırır. Bu bölümde, yan zincir işlemlerini yönetmek için bu üç temel yönetim modelinin nasıl kullanılacağı açıklanmaktadır.

### 5.3.1 Parlamento Yönetişim Modelini Kullanma

Parlamento modelinin yönettiği işler:

1. Yan zincir oluşturma  
Geliştiriciler özel bir yan zincir oluşturmak istediklerinde, bu yan zincir için bilişim desteği sağlamak için üretim düğümleri içerdiğinden parlamenter organizasyon tarafından ortak çözüm için bir başvuruda bulunmaları gerekir.
2. Yan zincir sözleşme dağıtımı/yükseltmesi  
Otomatik bir sözleşme incelemesi olmasına rağmen, güvenlik için, özel yan zincirin ve paylaşılmış yan zincir sözleşmesinin konuşlandırılması ve yükseltilmesinin ilk başlatılma sırasında parlamenter organizasyon tarafından onaylanması/dağıtılması gerekir. (Bu aynı zamanda yan zincir DApp yönetim sürecidir)
3. Bakiyede çapraz zincir indeksi ücret modelinin yönetimi/zamana dayalı ücret modelinin yönetimi  
Bakiye durumunda çapraz zincir indeksi ücret modeli/zamana dayalı ücret modeli, bu yan zincir için hizmet vermeye devam etmenin gerekip gerekmediğini belirlemek için teklif yönetimi yoluyla parlamenter organizasyon tarafından kararlaştırılır.

### 5.3.2 Birlik/Ortaklık Yönetişim Modelini Kullanma

Ortaklık, özerk bir kuruluştur ve ortaklık içindeki işler ortaklık yönetimi yoluyla ortaklaşa belirlenebilir. Örneğin:

1. Ortaklık üyeleri, hesap varlıklarını birlikte yönetir  
Ana zincir yönetimindeki "ortaklık üyeleri, hesap varlıklarını ortaklaşa yönetir" ile tutarlı.

### 5.3.3 Özelleştirilmiş Yönetişim Modeli Kullanma

Özel yönetim modeli kullanılarak yönetilen işlemler şunlardır:

1. Yan zincir blok indeks ücretinin parametrelerinin ayarlanması  
Çapraz zincir transferleri veya çapraz zincir doğrulaması elde etmek için ana zincirin yan zincirin bloklarını indekslemesi gerekir. Bu yan zincir için bir blok indeks ücreti ödenmelidir. Blok indeks ücreti, yan zincir geliştiricisi ile parlamenter organizasyon arasındaki konsültasyon yoluyla belirlenir.  
Ana zincir yönetimindeki Şekil 5.2.4-2'deki işlemler ile tutarlıdır.
2. Geliştirici ücreti modelinde önemli parametrelerin ayarlanması  
Ana zincir yönetimindeki "geliştirici ücret modelinde önemli parametrelerin ayarlanması" işlemiyle tutarlıdır.
3. Kullanıcı ücretlendirme modelinde önemli parametrelerin ayarlanması  
Ana zincir yönetimindeki "kullanıcı ücreti modelinde önemli parametrelerin ayarlanması" ile tutarlıdır.

Ayrıca Aelf'in özel yan zinciri "bir zincir, bir senaryo" için tasarlandığından geliştiriciler, senaryolarına uygun bir yönetim modeli oluşturmak için özel bir yönetim modeli kullanabilirler.