

Hybrid Cognitive System

Durum-Bilinçli, Kişilik Geliştiren ve Kapalı Döngülü Bilişsel Asistan Mimarisi

Mimari · Modül İç Mantıkları · Kapalı Döngü · Matematiksel Temeller · Doğrulama

Murat Yaşlıoğlu · 14 Eylül 2026 · Türkçe

İÇİNDEKİLER

Yönetici Özeti

1. Genel Mimari ve Tasarım Felsefesi

- 1.1 Tasarım Problemi
- 1.2 Dört Katmanlı Mimari
- 1.3 Tasarım Prensipleri

2. Pipeline Akışı

- 2.1 Konuşma Anı Akışı
- 2.2 Heartbeat Akışı

3. Modül Derinlemesine Analiz

- 3.1 Cognitive Watcher: Duygu / Ton / Bağlam Analizi
- 3.2 Cognitive Retention: Importance-Aware Hafıza
- 3.3 Affect: Boyutlu Duygu Modeli
- 3.4 Cognitive Inference Engine: Varlık / Plan / Çıkarım
- 3.5 Personality Development: Multi-Profile Kişilik
- 3.6 Attention Gate: Sosyal Zamanlama
- 3.7 Humor Policy: Mizah Politikası
- 3.8 Social Trainer: Sosyal Zeka ve Espri Kapanış Defteri
- 3.9 Metacognition: Üstbilis ve Güven Kalibrasyonu
- 3.10 Prospective Memory: İleriye Dönük Hafıza
- 3.11 Surprise → Curiosity Bridge: Merak Motoru
- 3.12 Memory Clusterer: Konu Bazlı Gruplama
- 3.13 Mood Journal: Ruh Hali Günlüğü
- 3.14 Guidance Feedback: Davranışsal Geri Bildirim Döngüsü
- 3.15 Memory Gist: Epizodik → Semantik Sıkıştırma
- 3.16 Self-Reflection: Haftalık Öz-Değerlendirme
- 3.17 Internal Narrative: Süren Benlik Anlatısı
- 3.18 Active Learning: Zayıf Alan Tespiti
- 3.19 Dreaming Loop: Hafıza Sentezi
- 3.20 Health Check: Sistem Sağlığı
- 3.21 Canlı Hattın Dışındaki Modüller

4. Veri Yapıları ve State Yönetimi

- 4.1 State dosyaları haritası
- 4.2 Atomik IO
- 4.3 Retention state yapısı

5. Algoritmalar ve Matematiksel Temeller

- 5.1 Önem skoru (importance)
- 5.2 Hibrit geri getirme (RRF)
- 5.3 Affect dinamiği
- 5.4 Trait güncelleme matematiği
- 5.5 Mizah kalibrasyonu: Beta-Bernoulli
- 5.6 Güven kalibrasyonu
- 5.7 Hafıza zayıflaması

6. Modüller Arası Veri Akışı

- 6.1 Ana veri akış şeması
- 6.2 Modül bağımlılık özeti
- 6.3 Kritik veri yolları

7. Doğrulama ve Ölçüm Dürüstlüğü

- 7.1 Regresyon testleri
- 7.2 Sağlık denetimi
- 7.3 Göç betikleri
- 7.4 İzlenen hata sınıfları
- 7.5 Doğrulama rejiminin göremedikleri

8. Sınırlılıklar ve Gelecek Çalışma

- 8.1 Doğrulanmamış iddialar
- 8.2 Yapısal sınırlar
- 8.3 Öncelikli gelecek çalışma
- 8.4 Kapanış değerlendirmesi

Ek A: Komut Referansı

Ek B: Terminoloji

Bu Belge Hakkında

Bu metin, Hybrid Cognitive System (HCS) mimarisinin bütünsel bir teknik tarifidir. Sistemin katman yapısını, yirmi canlı modülünün iç mantığını, modüller arası veri akışını, kullanılan veri yapılarını ve matematiksel temellerini kayda geçirir.

Metnin yazımında iki ilke gözetilmiştir. Birincisi, mimarinin yalnızca tasarım niyetiyle değil ölçülmüş davranışıyla tarif edilmesi: her sayısal değer canlı durum dosyalarından okunmuştur. İkincisi, doğrulanmamış iddiaların açıkça sınırlılık olarak işaretlenmesi; bu iddialar Bölüm 8’de toplu halde listelenmiştir.

Yönetici Özeti

Hybrid Cognitive System, geleneksel konuşma asistanlarının üç temel kısıtını (hafızasızlık, kişiliksizlik ve bağlamsızlık) aşmak üzere tasarlanmış, durum-bilinçli (stateful) ve kişilik geliştiren bir bilişsel mimaridir. Sistem, her konuşmayı sıfırdan başlatan klasik yaklaşımın aksine, kalıcı bir hafıza katmanı, sürekli güncellenen bir kişilik vektörü ve karşı tarafın duygusal/bağlamsal durumunu okuyan bir izleme katmanı üzerine kurulur.

Mimarinin tanımlayıcı özelliği kapalı döngü olmasıdır. Bilişsel sistemin ürettiği tüm sinyaller (bağlamsal olarak modüle edilmiş trait vektörü, boyutlu ruh hali, ton rehberi, mizah direktifi, ateşlenen hatırlatmalar ve geri getirilen anılar) tek bir metin bloğunda (persona_block) toplanır ve yanıt üretiminden hemen önce dil modelinin bağlamına yazılır. Bu enjeksiyon adımı, sistemi bir gözlemciden bir aktöre dönüştüren yapısal karardır: ölçüm ile davranış arasındaki bağ burada kurulur. Enjeksiyon katmanı olmadan sistem bir günlük tutar ama günlüğü hiç açmaz.

Mimari dört zamansal katmana ayrılır. Konuşma anı katmanı, yanıtı anlık olarak biçimlendiren çekirdek hattıdır. Heartbeat katmanı, yaklaşık 30 dakikalık aralıklarla devreye giren ve sağlık denetimi, hafıza zayıflaması, kümeleme ile rüya sentezini yürüten arka plan döngüsüdür. Haftalık katman öz-değerlendirme, epizodik-semantik sıkıştırma ve iç anlatının yeniden yazımını içerir. Enjeksiyon katmanı ise diğer üçünün çıktısını yanıt üretimine bağlar.

Bu belge, sistemin 20 canlı modülünü tek tek ele alır; her modülün çalışma prensibini, veri yapılarını, matematiksel formüllerini ve modüller arası bağımlılıklarını sunar. Amaç, hem mühendislik düzeyinde yeniden üretilebilir bir referans hem de örgütsel davranış ile bilişsel/davranışsal bilimler literatürüne bağlanabilecek kavramsal bir zemin sunmaktır.

Okuma Rehberi

Bölüm 1–2 mimariyi ve pipeline akışını üst düzeyde anlatır; sistemin nasıl “düşündüğünü” kavramak için buradan başlanmalıdır.

Bölüm 3 her modülün iç mantığını ayrıntılandırır ve belgenin gövdesini oluşturur.

Bölüm 4–5 veri yapılarını ve matematiksel temelleri verir; yeniden üretim (replication) için kritik referanstır.

Bölüm 6 modüller arası veri akışını ve bağımlılık matrisini gösterir.

Bölüm 7 doğrulama rejimini ve ölçüm dürüstlüğü ilkesinin operasyonel karşılığını belgeler.

Bölüm 8 sınırlılıkları ve gelecek çalışma yönlerini dürüstçe sıralar.

1. Genel Mimari ve Tasarım Felsefesi

1.1 Tasarım Problemi

Geleneksel yapay zeka asistanları durumsuzdur (stateless): her konuşmayı sıfırdan başlatır, geçmişini hatırlamaz ve zamanla bir kişilik geliştirmeyebilir. Bu, kullanıcı açısından her seferinde “tanışılan” ama hiçbir zaman “tanıyan” bir muhatapla konuşma deneyimi yaratır. Hybrid Cognitive System tam da bu sınırı aşmak için tasarlanmıştır. Aşağıdaki tablo, mimarinin çözmeyi üstlendiği problemleri ve her birinin yapısal karşılığını gösterir.

PROBLEM	MİMARİ ÇÖZÜM	DAVRANIŞA BAĞLANMA BİÇİMİ
Hafızasızlık	Retention: decay-aware kalıcı anı deposu	Hibrit geri getirme ile her mesajda prompt’a girer
Kişiliksizlik	Multi-profile, 17 trait’lik kişilik vektörü	Setpoint dinamiği + bağlamsal trait aktivasyonu
Bağlamsızlık	Watcher → duygu / ton / ironi analizi	Boyutlu affect modeli (valence × arousal)
Proaktif olamama	Inference engine → çıkarım → proaktif soru	Sürekli ve tekilleştirilmiş soru kuyruğu
“Şimdi konuşma zamanı mı?”	Attention Gate → alıcılık skoru	Ton rehberi ve mizah politikasına girdi
Öğrenememe	Social Trainer + Metacognition	Esprî kapanış defteri + Brier kalibrasyonu
Sessiz çürüme	Health Check → modül canlılık denetimi	12 metrikli kontrol, sessiz bozulmaları yakalar

Bu çözümlerin ortak teması, sistemin zamanı bir değişken olarak içselleştirmesidir. Hafıza zamanla zayıflar, kişilik zamanla mizaç temeline döner, alıcılık günün saatine göre değişir ve çıkarımların güvenilirliği geçmiş isabet oranlarına göre yeniden kalibre edilir. Sistem, bu nedenle statik bir kural tablosu değil, sürekli yeniden dengelenen dinamik bir yapıdır.

1.2 Dört Katmanlı Mimari

Sistemin çalışması, farklı zaman ölçeklerinde işleyen katmanlara ayrılır. Bu ayrım, hem hesaplama maliyetini yönetmek hem de insan bilişindeki “anlık tepki, gün içi pekiştirme, uzun dönem bütünleştirme” ayrımını taklit etmek için kurgulanmıştır.

ENJEKSİYON (yanıttan ÖNCE)
persona_block → prompt
trait + ruh hali + ton rehberi + mizah + recall
KONUŞMA ANI (her mesaj)
Watcher → Affect → Retention + Recall → Inference
→ Attention Gate → Humor Policy → Espri kapanışı
→ Personality → Prospective / Surprise / Guidance
HEARTBEAT (~30 dk, hafif)
Health Check → Decay → Cluster → Dreaming
→ Niyet kontrolü → Merak keşfi
HAFTALIK (Pazar)
Self-Reflection → Memory Gist → Internal Narrative

Enjeksiyon katmanı mimarinin kilit taşıdır. Diğer üç katmanın ürettiği tüm durum, yanıt üretiminden hemen önce tek bir metin bloğuna dönüştürülür ve dil modelinin bağlamına yazılır. Bilişsel makinenin ürettiği hiçbir sinyal, bu katmandan geçmeden davranışa dönüşemez.

Konuşma anı katmanı mesaj geldiği anda tetiklenir ve on dört adımlı bir hat olarak işler: bağlam kurma, geri getirme, çıkarım, sosyal zamanlama, mizah kararı, enjeksiyon, yanıt ve yanıt sonrası öğrenme. Kişilik güncellemesi ayrı bir süreç değil, aynı hattın son adımıdır.

Heartbeat katmanı yaklaşık 30 dakikada bir “poll” eder; önce sistem sağlığını denetler, ardından hafızayı zayıflatır, anıları kümeler ve rüya döngüsünü çalıştırır. Maliyetli işler sayaç tabanlı koşullarla seyrekleştirilir.

Haftalık katman en ağır bütünleştirme işlerini yürütür: öz-değerlendirme, epizodik özlerin çıkarımı ve iç anlatının yeniden yazımı.

1.3 Tasarım Prensipleri

- **İnteraktif canlılık:** Cron yoktur. Sistem yalnızca konuşma anında ve heartbeat poll’unda çalışır. Gece saatlerinde (23–08) sessiz moda geçer.
- **Modüler bağımsızlık:** Her modül bağımsız yüklenir; biri çökse bile diğerleri çalışmaya devam eder. Yüklenme durumu, dağınık try/except bayrakları yerine tek bir modül registry üzerinden yönetilir; hangi modülün canlı olduğu tek komutla görülebilir.
- **Atomik IO:** Tüm state yazımları `atomic_json_dump()` ile yapılır; çökme anında dosya bozulması önlenir.
- **Tek otorite:** Her kavramın tek bir tanım yeri vardır. Duygu ölçekleri `affect.py`’de, `trait` dinamiği `personality_development.py`’de, mizah kararı `humor_policy.py`’dedir. Aynı kavram için birden fazla tablo tutulmaz.
- **Ölçüm dürüstlüğü:** Hiçbir istatistik örneklem büyüklüğü olmadan raporlanmaz. $n < 5$ ise oran yerine “yetersiz veri” döner. Sahte metrik üretmek yasaktır; veri yoksa döngü atlanır.
- **Test edilebilirlik:** Düzeltilen her hata için bir regresyon testi yazılır. Test dosyası, hangi hatanın korunduğunu docstring olarak taşır.

- Denetlenebilirlik: Her olgu ve çıkarım, dayandığı öncüllerle (premises) birlikte saklanır; her karar insan-okunur bir gerekçeye bağlanabilir.

Tasarım Notu: Neden cron değil heartbeat?

Cron tabanlı bir sistem, kullanıcı orada olmasa bile sabit aralıklarla çalışır ve kaynak tüketir. Heartbeat + konuşma-anı modeli ise sistemin “canlılığını” etkileşime bağlar: sistem yalnızca bir muhatap varken ve ona yakın zamanda dokunulduğunda uyanık kalır. Bu, hem enerji/maliyet açısından ekonomik hem de davranışsal olarak daha doğal bir mevcudiyet hissi üretir.

Bu tercihin bir maliyeti vardır: etkileşime bağlı bir sistem, etkileşim kesildiğinde donar ve sessizlik dönemleri boyunca durum hiç sönümlenmezse sistem, günler öncesinin ruh haliyle uyanır. Mimari bu maliyeti zaman-bazlı gerileme (time regression) ile karşılar: hem ruh hali hem trait vektörü, geçen süreye göre temele doğru çekilir. Sessizlik de bir sinyaldir.

2. Pipeline Akışı

2.1 Konuşma Anı Akışı

Hat, üretilen tüm sinyalleri toplayan bir enjeksiyon adımıyla kapanır. Aşağıdaki şema tam sırayı gösterir; [E] işaretli adım, yanıt üretiminden önce prompt'a yazılan bloktur.

Muhataap mesaj atar

- [0] normalize_text() - tipografik kesme, NBSP, Türkçe I/İ
- [1] Watcher.on_message(text, sender_id)
duygu · yoğunluk · sarkazm · situation · is_joke
humor_score · laugh_strength
- [2] Affect.observe(sender, emotion, intensity, situation)
valence/arousal · üç zaman ölçeği · geçiş sürprizi
- [3] Retention.store(text, role="user", emotion, intensity)
importance (toplamsal ağırlıklı) · O(1) dedup
- [4] Retention.recall_block(text, k=3) ← KRİTİK
hibrit arama (TF-IDF + BM25 → RRF) + spaced reminder
+ cross-session köprüler
- [5] InferenceEngine.process_message(text)
varlık · olgu · plan · çıkarım (metaboliş kaydı içeride)
- [6] Prospective.check(fire=False) · Surprise.observe()
· GuidanceFeedback.on_message()
- [7] AttentionGate.assess(sender) → receptivity · guidance
- [8] SocialTrainer.close_pending_jokes(text)
önceki espri tuttu mu? (kahkaha şiddetiyle)
- [9] HumorPolicy.assess() → seviye · alt tip · callback
- [E] persona_block üretilir ve prompt'a yazılır

ŞU ANKİ HALİM
Baskın: loyalty .84, warmth .78, ...
Muhatabın ruh hali: normal (+0.00, durgun)
Ton: enerjik, samimi · uzunluk: normal
MİZAH: normal (0.51) · sarkazm: kapalı
İlgili anılar: (34g önce) ...
HATIRLAT: ...
- [10] Yanıt üretilir (dil modeli)
- [11] Watcher.on_message(yanıt, sender_id="hcs")
sistemin kendi duygusu etiketlenir; espri yapıldıysa
SocialTrainer.open_joke() ile bekleyen kayıt açılır
- [12] Retention.store(yanıt, role="assistant", gerçek duygu)
- [13] Personality.update_from_watcher() + update_from_event()
habituation · setpoint decay · zaman gerilemesi

Akışın Mantığı

Bağlam-kurucu modüller (Watcher, Affect, Retention, Recall, Inference) yanıttan önce; öğrenme ve kişilik modülleri ise yanıttan sonra çalışır. Bu sıralama, yanıtın en güncel bağlamla üretilmesini, kişilik ve alıcılık güncellemelerinin ise bu etkileşimin sonucundan beslenmesini sağlar.

Hatta iki gecikmeli ölçüm döngüsü vardır ve ikisi de aynı deseni kullanır: Guidance Feedback (record_issued → on_message) ve espri kapanış defteri (open_joke → close_pending_jokes). Her ikisinde de bir müdahalenin etkisi ancak bir sonraki mesajda ölçülebilir; ölçüm anı ile karar anı bilinçli olarak ayrılmıştır.

2.2 Heartbeat Akışı

Heartbeat, sistemin arka plan bakım döngüsüdür. Maliyetli işleri seyrek tetiklemek için sayaç tabanlı koşullar kullanır. Döngü, sessiz bozulmaları erken yakalamak için sağlık kontrolüyle başlar.

```
Sistem poll eder (~30 dk)
|
| health_check.py --json (her heartbeat)
|   hafıza bütünlüğü · sessiz çürüme · runtime köprüsü
|
| cognitive_retention.py --decay-check (günde 1)
|   importance kovalasına göre yaşlandırma
|
| memory_clusterer.py --run (5 heartbeat'te 1)
|
| dreaming_loop.py --cycle 1 (10'da 1, 08-23 arası)
|   semantik embedder ile bağlantı arama
|
| prospective_memory.py --check (her heartbeat)
|   → ilet → --ack <id> (iki aşamalı teslim)
|
| surprise_curiosity.py --pop (10'da 1)
|   → araştır → --finding <id> "<öğrenilen>"
```

İki aşamalı teslim deseni burada kritiktir: check() yalnızca tetiklenen niyetleri listeler, kaydı kapatan acknowledge(id) çağrısıdır. Böylece iletilmemiş bir hatırlatma bir sonraki döngüde yeniden tetiklenir; teslim edilmeden “yapıldı” işaretlenen niyet kalmaz. Aynı desen merak kuyruğunda da uygulanır: pop() bir konuyu alır, record_finding() ne öğrenildiğini kaydeder ve halkayı kapatır.

3. Modül Derinlemesine Analiz

Bu bölüm sistemin 20 canlı modülünü ele alır. Her modül için çalışma prensibi, algoritma adımları ve veri yapıları sunulur. Canlı hattan çıkarılmış dokuz modül ise Bölüm 3.21’de ayrıca ele alınır.

3.1 Cognitive Watcher: Duygu / Ton / Bağlam Analizi

Pattern-matching tabanlı, <100 ms hedefli bir analiz katmanıdır. Bir makine öğrenmesi modeli değildir; Türkçe’ye özel bir lexicon ile emoji, noktalama ve bağlam kurallarını birleştirerek çalışır. Bu tasarım tercihi hız ve yorumlanabilirlik lehine bilinçli bir dengedir: model her kararını insan-okunur kurallara dayandırır.

ALGORİTMA ADIMLARI

1. Metin temizleme ve normalizasyon (URL, mention; tipografik kesme → ASCII).
2. Emoji analizi: her emoji için duygu + yoğunluk eşleştirmesi; tekrar sayısı yoğunluğu yükseltir.
3. Kelime bazlı duygu skorlaması: pozitif, negatif, sarkazm ve Türkçe internet argosu sözlükleri.
4. Çoklu duygu skorlaması: her kategori için ayrı toplam; en yüksek olan birincil duygu olur.
5. Sarkazm/ironi tespiti: çoklu-indikatör bonuslu doygun formül; sarkazm yoksa “sarcastic” etiketi düşürülür.
6. Mizah tespiti: humor_score, humor_subtype ve laugh_strength.
7. Durum (situation) sınıflandırması ve bağlamsal ipuçları (is_serious, is_family, is_technical...).

Gülme yazımı, Türkçe sohbet dilinde kelime sınırı varsayımını bozan bir alandır: “ahahah”, “hahahaha” gibi biçimler sabit kalıplarla yakalanamaz. Bu nedenle tespit, tekrarlı-hece düzenli ifadesine dayanır: (?a?h[ae]){2,}h?. Yumuşak gülümseme emojileri de (😊 😊 😊 😊) mizah işaretçileri arasındadır; aksi halde samimi takılmalar hakaret olarak sınıflanır.

KAHKAHA ŞİDDETİ (LAUGH_STRENGTH)

İkili “tuttu / tutmadı” sinyali, nazik bir gülümseme ile gerçek bir kahkahayı aynı sayar. Bu nedenle tepki gücü sürekli bir ölçekte hesaplanır ve mizah kalibrasyonuna ağırlıklı gözlem olarak girer.

😊 tek	= 0.60	ölüyorum/koptum/yıkıldım	= 1.00
😊😊😊	= 0.90	💀	= 0.85
"ahahahaha"	= $\min(0.30 + 0.08 \cdot \text{uzunluk}, 1.0)$		
😊 / 😊	= 0.30	teпки yok	= 0.00

SAMİMİYET-HAKARET AYRIŞTIRICI

Türkçede yakın ilişkide kullanılan “oğlum”, “kardeşim”, “reis” gibi hitaplar ile “mal mısın”, “deli misin” gibi ifadeler bir hakaret değil, samimiyet göstergesidir. Bunları saldırı sinyali olarak kaydeden bir sistem, muhatabı şakalaştıkça içine kapanır. Bu nedenle hitap veya pozitif emoji varlığında saldırı şiddeti 0.2 ile çarpılır ve sinyal “playful_jab” olarak yeniden etiketlenir.

3.2 Cognitive Retention: Importance-Aware Hafıza

Anılar, ham metin ve metadata olmak üzere iki paralel dizide tutulur. İki dizi arasındaki eşleme konumsal değil, kimlik (idx) tabanlıdır; erişim yalnızca bir idx sözlüğü üzerinden yapılır ve sözlük tabanlı bir indeks önbellesi kullanılır. Konumsal eşleme, dizilerin uzunlukları herhangi bir nedenle ayrıştığında importance ve duygu etiketlerinin başka bir anıya bağlanmasına yol açar; kimlik tabanlı erişim bu sınıf hatayı yapısal olarak imkansız kılar.

```
conversation_memory[N]      ↔  memory_metadata[N]  (idx ile eşlenir)
"2026-07-11 11:07: [user] ..."  { idx, emotion, intensity,
                                     importance, role, timestamp,
                                     review_count, decay_state }
```

DECAY STRATEJİSİ

Anılar önem düzeylerine göre dört kovaya ayrılır; her kova farklı bir saklama–zayıflama–silme takvimine tabidir. Bu, insan hafızasındaki “önemli olan kalır, sıradan olan unutulur” ilkesinin operasyonel karşılığıdır.

IMPORTANCE ARALIĞI	TAM SAKLAMA	ZAYIFLAMA	SİLME
0.0 – 0.3 (düşük)	14 gün	30 gün	60 gün
0.3 – 0.6 (orta)	30 gün	60 gün	120 gün
0.6 – 0.8 (önemli)	60 gün	120 gün	365 gün
0.8 – 1.0 (kritik)	90 gün	180 gün	730 gün

Decay işlemi metni hiçbir koşulda değiştirmez. Zayıflama ve arşivleme durumu yalnızca metadata’da (decay_state) tutulur; etiketleme sunum katmanında yapılır. Metnin fiziksel olarak kırılması iki geri dönüşsüz sonuç doğurur: en eski ve çoğu zaman en değerli kayıtlarda veri kaybı ve tarih ayrıştırmasının bozulması nedeniyle bir kez arşivlenen kaydın bir daha hiçbir bakım döngüsüne girememesi. Metin dokunulmaz tutularak her iki risk de ortadan kaldırılmıştır.

HİBRİT GERİ GETİRME

Geri getirme, ardışık bir yedekleme (fallback) değil gerçek bir birleştirmedir. Semantik ve anahtar kelime yolları paralel çalışır, iki aday listesi Reciprocal Rank Fusion ile birleştirilir ve sonuç önem/tazelik ile yeniden sıralanır. Ardışık yedekleme tasarımında tek bir zayıf semantik isabet, çok daha iyi eşleşecek anahtar kelime yolunu tamamen susturur; RRF bu asimetriyi ortadan kaldırır. Ayrıntılı formül için bkz. Bölüm 5.2.

TEKRAR TURU KORUMASI VE ROL AĞIRLIKLILIK ÖNEM

Model aynı turu birden fazla kez --pair ile gönderebiliyor; ölçülen bir örnekte tek bir mesaj art arda dört çağrı üretti ve her çağrı ayrı bir anı, kişilik olayı ve çıkarım kaydı açtı. Bu nedenle 10 dakikalık bir pencere içinde aynı rol ve normalize edilmiş metinle eşleşen bir tur artık yeni kayıt açmaz; yalnızca asistan cevabı yerinde

güncellenir. Asistan tarafından üretilen kayıtların önem skoru ayrıca 0.7 ile çarpılır: duygu ve emoji yüklü asistan cevapları, kullanıcının kısa mesajlarını önem sıralamasında geride bırakıyordu (ölçülen ortanca: asistan ~0.56, kullanıcı ~0.30).

3.3 Affect: Boyutlu Duygu Modeli

Duyguyu etiket sayacı olarak tutan bir tasarımın (happy×n, angry×m) üç yapısal sorunu vardır. Birincisi, duygu ataleti yoktur; tek bir mesaj ruh halini sıfırdan tavana taşıyabilir. İkincisi, anlık duygu ile ruh hali ayrımı yoktur; kalıcı bir ruh hali durumu tutulmaz. Üçüncüsü, yoğunluk ölçekleri modülden modüle tutarsızlaşır.

Affect modülü bu üç sorunu birlikte çözer: her mesajı bir (valence, arousal) noktasına eşler ve üç zaman ölçeğinde üstel yumuşatma uygular. Ayrıca duygu ölçeklerinin tek otoritesidir; diğer tüm modüller ölçeklerini buradan alır.

KATMAN	A	YAKLAŞIK YARI ÖMÜR	ANLAMI
emotion	0.50	~1 mesaj	Şu anki tepki
mood	0.05	~14 mesaj / 36 saat	Son saatlerin ruh hali
temperament	-	trait decay (gün-hafta)	Mizaç (personality)

Modül ayrıca üç mekanizma daha sağlar: sessizlikte ruh halinin temele dönmesi (zaman gerilemesi), tekrarlanan uyarana duyarsızlaşma (habituation) ve duygu geçiş matrisi üzerinden beklenmediklik ölçümü. Sonuncusu merak motoruna sürpriz sinyali verir.

3.4 Cognitive Inference Engine: Varlık / Plan / Çıkarım

Bir bilgi grafiği (KnowledgeGraph) üzerinde dört temel yapı taşınır: varlıklar düğümleri, olgular kenarları, planlar zamansal niyetleri ve çıkarımlar türetilmiş sonuçları temsil eder. Motor tamamen deterministiktir; hiçbir dil modeli çağrısı yapmaz. Her olgu ve çıkarım, dayandığı öncüllerle birlikte saklanır; bu, denetlenebilirlik açısından bilinçli bir tercihtir.

TÜR	MANTIK	ÖRNEK
Temporal	Plan tarihi geçti mi?	"1 hafta sonra Tokyo" + 10 gün → "muhatap Tokyo'da"
Spatial	Hiyerarşik konum	Berlin → Almanya, Asakusa → Tokyo
Relational	Kişi ilişkileri	eş, çocuk, meslektaş bağları
Behavioral	Tekrar eden örüntüler	"Her sabah kahve sorar"

AYRIŞTIRMA VE YAŞAM DÖNGÜSÜ KURALLARI

Deterministik bir çıkarım motorunun isabeti, tamamen kalıpların dayanıklılığına bağlıdır. Motorda beş kural bu dayanıklılığı sağlar:

- Tipografik normalizasyon. Kalıplar yalnızca ASCII apostrofu (U+0027) kabul edemez; iOS/macOS klavyeleri ve mesajlaşma istemcileri varsayılan olarak U+2019 üretir. Normalizasyon adımı [0], telefonda yazılan cümlelerin sessizce ayrıştırılamamasını önler.
- Kelime sınırlı lokasyon eşleşmesi. Saf substring eşleşmesi Türkçede yüksek yanlış-pozitif üretir (“usa” → Asakusa, “tr” → trafik). Eşleşme kelime sınırlıdır; üç harften kısa kısaltmalarda iki taraflı sınır şartı aranır.
- Adlandırılmış yakalama grupları. Kalıp ile işleyici arasında grup sayısına dayalı örtük sözleşme kurulmaz; gruplar adlandırılır, böylece kalıp eşleştiği halde hiçbir şey yazılmaması gibi sessiz kopmalar oluşmaz.
- Olguların yaşlanması. Yinelenen olgu birleştirmesinde güven yalnızca yukarı gidemez; olgular 14 günlük yarı ömre tabidir ve tekil-değerli yüklem için “supersede” bayrağı tutulur.
- Planların bayatlaması. Planların sona erme koşulu vardır (30 günlük eşik); aksi halde bir kez etkin olan plan sonsuza dek etkin kalır ve sistem var olmayan bir seyahat hakkında proaktif soru sorar.

3.5 Personality Development: Multi-Profile Kişilik

Kişilik, 17 trait’ten oluşan bir vektördür ve her trait kendi zayıflama hızına sahiptir. Bu, bazı özelliklerin (merak) hızla değişebildiği, bazılarının (sadakət) çok yavaş kaydığı bir kişilik kararlılığı profili yaratır.

KATEGORİ	TRAIT'LER	DECAY RATE
Sosyal / duygusal	warmth, empathy, loyalty, playfulness, mischief, humor_style	0.001 – 0.005
Bilişsel	assertiveness, wisdom, analytical, curiosity, technical_depth, creativity, thoroughness, patience, introspection	0.001 – 0.006
Genişletilmiş	protectiveness, adventurousness	0.003 – 0.006

MULTİ-PROFILE MİMARİSİ

Sistem her muhatap için ayrı bir trait seti tutar. Aynı “kişi” farklı kişilere karşı farklı tonlarda davranabilir; bu, insan ilişkilerindeki bağlama-özgü benlik sunumunun karşılığıdır. Profil tipleri ilişki türüne göre ayrışır: “spouse” ve “child” aynı aile profilini paylaşmaz, çünkü iki ilişkinin ton, koruyuculuk ve mizah beklentileri belirgin biçimde farklıdır.

Profil anahtarı tek bir kuralla üretilir: key = sender_id. Anahtar üretiminin koşullu dallara bağlanması, aynı muhatap için birden fazla profilin sessizce oluşmasına ve haftalık raporların ölü bir profilden okumasına yol açar; tek kural bu riski yapısal olarak kapatır.

GÜNCELLEME MANTIĞI

1. Watcher sonucunu al: emotion, intensity, situation.
2. Zaman gerilemesi: son etkileşimden bu yana geçen süreye göre trait'leri mizaç temeline doğru çek.
3. Durum deltalarını hesapla (SITUATION_TRAIT_MAP), yoğunluk çarpanı ve güven ağırlığı uygula, ± 0.03 ile kırp.
4. Konu modülasyonu ekle: kelime sınırlı eşleşme ve ayrı, daha küçük bir tavan (± 0.012) ile.
5. Habituation: aynı durum son beş etkileşimde kaç kez tekrarlandıysa deltayı $1/(1+0.3 \cdot n)$ ile sönümle.
6. Setpoint decay: her trait'i mizaç temeline doğru çek (mutlak azalma değil).
7. Deltaları uygula ve $[0.05, 1.0]$ aralığına kırp.

Mutlak azalma yerine bir çekim merkezine dönüş kullanılması, trait'in ne tavana yapışmasına ne tabana çökmesine izin verir. Doygun bir trait ayırt edici sinyal taşımaz: “çok sıcak” ile “aşırı sıcak” arasında fark kalmaz ve kişilik vektörü karar için kullanılamaz hale gelir. Setpoint modeli ayrıca “kişilik değişir ama özü kalır” semantiğini doğrudan modele sokar; mizaç temeli, profil tipinin başlangıç değerleri olarak kalıcı biçimde saklanır.

3.6 Attention Gate: Sosyal Zamanlama

“Şu an konuşmak için iyi bir zaman mı?” sorusuna nicel bir yanıt üretir. Karşı tarafın son dönemdeki duygusal valansını ve günün saatini birleştirerek bir alıcılık (receptivity) skoru hesaplar. Sistemin en olgun ve en uzun süredir kesintisiz çalışan geri bildirim halkasıdır.

ALICILIK HESAPLAMA

1. Son 90 dakikadaki watcher kayıtlarını al (en fazla 12 mesaj).
2. Her mesajın duygusal valansını yoğunlukla ağırlıklandır.
3. Zamana göre ağırlıklı ortalama al (yeniler daha belirleyici).
4. Taban 0.6'dan başla; valansa göre ± 0.3 , negatif sinyalde -0.35 , sessiz saatte -0.4 , ciddi konuda -0.15 uygula.
5. user_state sınıflandır: stressed / upset / busy / engaged / relaxed / quiet_hours.

```
{ "user_state": "engaged", "receptivity": 0.70,  
  "guidance": { "tone": "enerjik, samimi", "brevity": "normal",  
                "proactive_ok": true, "max_proactive_questions": 2 } }
```

3.7 Humor Policy: Mizah Politikası

Mizahın ölçüm tarafı (espri tespiti, alt tip sınıflandırması, mizah trait'leri) tek başına davranış üretmez; ölçen ama karar vermeyen bir sistem termometredir, termostat değildir. Humor Policy bu boşluğu kapatan karar katmanıdır.

Girdileri Attention Gate'in alıcılık skoru, mizah kalibrasyonunun Beta posterior'ları, aktif profilin playfulness/mischief/humor_style değerleri ve muhatabın ruh halidir. Çıktısı prompt'a yazılan Türkçe bir direktiftir.

MİZAH: normal (0.51)
neden: ciddi konu
tutan konular: kahve, Galatasaray
kaçın: hukuk, sağlık
sarkazm: kapalı · tercih: pun, absurd, general
(veri az: n=1.6; kalibrasyona güvenme, deneyerek öğren)
🔄 geri dönüş fırsatı: "Kahvesiz kod yazmak..." (konu: kahve)

ZAMANLAMA KAPILARI

Mizahın yarısı zamanlamadır. Politika, öğrenilmiş temel seviyeyi dört kapıdan geçirir: ciddi konu (-%45), gece saati (-%50), düşük alıcılık (-%40) ve kötü ruh hali (-%30). Hassas konularda (hukuk, sağlık, dava) seviye doğrudan sıfırlanır.

RUH HALİNE GÖRE MİZAH TÜRÜ

Seviye ile tür ayrı kararlardır. Muhatap çökkün ya da gerginse alaycı ve kara mizah bastırılır; kendi üstünden espri ve sıcak mizah öne çıkarılır. Kendi üstünden espri güven verir, alaycı espri yaralar. Bu ayrım sabit bir tabloyla değil, ruh hali çeyreğine göre ağırlıklandırma ile yapılır.

3.8 Social Trainer: Sosyal Zeka ve Espri Kapanış Defteri

Social Trainer, hangi tonun hangi muhatapta işe yaradığını öğrenen pekiştirme tabanlı bir katmandır. Mizah tarafındaki tasarımın kilit noktası, ölçülen sinyalin yönüdür: gelen mesajdaki is_joke bayrağı “muhatap espri yaptı” demektir, “sistemin esprisi tuttu” demek değildir. Kalibrasyon için gereken sinyal, sistemin kendi esprisine gelen tepkidir; bu nedenle ölçüm gecikmeli bir defter üzerinden yapılır.

KAPANIŞ DEFTERİ (PENDING JOKE LEDGER)

Sistem bir espri yaptığında bekleyen bir kayıt açılır; muhatapın bir sonraki mesajı bu kaydı kapatır. Desen, Guidance Feedback’in record_issued → on_message deseninin birebir kopyasıdır; kanıtlanmış bir mekanizmanın yeni bir alana taşınması, sıfırdan tasarımdan daha güvenilir bir yoldur.

KOŞUL	SONUÇ	AĞIRLIK
$\Delta t < 180$ sn ve gülme sinyali	landed	laugh_strength (0.3 – 1.0)
$\Delta t < 180$ sn ve soğuk emoji / negatif sinyal	bombed	-1.0
$\Delta t > 600$ sn veya tepki yok	ignored	-0.4 (zayıf negatif)

SOĞUMA BÜTÇESİ VE GERİ DÖNEN ESPRİ

Bir espri tutmazsa üç tur mizah kapatılır; görmezden gelinirse bir tur. Tutan espri sayacı sıfırlar. Bu, insan mizah davranışının temel dinamiğidir: ortam soğuduysa zorlamamak. Ayrıca tutan espriler arşivlenir; yeni mesaj eski bir espriyle konu örtüşmesi gösterirse “geri dönüş fırsatı” sinyali üretilir. Geri dönen espri (callback humor), hafızası olan bir dostun yapabileceği, tek başına bir dil modelinin yapamayacağı şeydir.

3.9 Metacognition: Üstbilış ve Güven Kalibrasyonu

Üstbilış katmanı, sistemin kendi çıkarımları hakkında düşünmesini sağlar: her tahmini açıkça kaydeder, sonradan doğrulanıp doğrulanmadığını izler ve buna göre gelecekteki güvenini yeniden kalibre eder.

Kalibrasyonda az veri, bir eşik kapısıyla değil Laplace düzeltmesiyle yönetilir: veri azken çarpan kendiliğinden 1.0'a yakınsar, veri geldikçe gerçek orana yaklaşır. Sabit bir “ $n < 3$ ise kalibrasyon yapma” kapısı, yalnızca çözülmüş tahminlerle beslenen bir sayaçla birleştğinde kalibrasyonu fiilen kalıcı olarak kapatır; düzeltme temelli yaklaşım bu tuzağı barındırmaz.

İkinci bir tasarım kararı, kalibrasyonun keskinlik (sharpness) boyutunu da ölçmektir. Yalnızca isabet/ortalama güven oranına bakan bir formül, 0.9 güvenle %90 isabet ile 0.5 güvenle %50 isabeti aynı sayar; oysa bu ikisi çok farklı bilgi değeri taşır. Bu nedenle Brier skoru ve güvenilirlik kovaları tutulur: sistem artık “temporal tipte 0.8 güvenle %45 isabet” diyebilir.

3.10 Prospective Memory: İleriye Dönük Hafıza

İleriye dönük hafıza, gelecekte bir şeyi yapmayı hatırlama yetisidir. Sistem bunu iki tetik mekanizmasıyla uygular: tarih tetiği ve konu tetiği.

Niyet yakalamada belirleyici olan, kalıp listesinin kapsamıdır. Yalnızca açık ifadelerden (hatırlat, unutma, aklında olsun, not al, takip et) oluşan bir liste Türkçede pratik olarak boş döner; çünkü niyet çoğunlukla örtük kurulur: “aramam lazım”, “yapmalıyım”, “söz verdim”. Kalıp kümesi bu örtük biçimleri kapsayacak şekilde tanımlanmıştır.

Teslim iki aşamalıdır: check() yalnızca tetiklenen niyetleri listeler, acknowledge(id) kaydı kapatır. Tetiklenen niyetin anında “fired” yazılması, ajanın o turda hatırlatmayı iletmemesi halinde sessiz kayba yol açar: niyet bir daha hiç tetiklenmez. İki aşamalı teslim bu riski ortadan kaldırır.

Türkçe tarih ayrıştırmasında ayrı bir sınır kuralı uygulanır: sözlük uzundan kısaya taranır ve iki taraflı sınır aranır. Aksi halde “cuma” ile “cumartesi”, “pazar” ile “pazartesi” çiftlerinde kısa olan uzun olanı yutar ve randevu yanlış güne yazılır.

3.11 Surprise → Curiosity Bridge: Merak Motoru

Bu modül beklenmedikliği bir öğrenme tetikleyicisine dönüştürür: yeni veya nadir kelimeler içeren mesajlar yeni konu olarak işaretlenir ve merak kuyruğuna eklenir. Heartbeat sırasında bu konular araştırılır. Kuyruk bir “yapıldı” listesi değil, kapanan bir öğrenme halkasıdır: pop() ile alınan konu, record_finding() ile ne öğrenildiği kaydedilerek kapatılır.

3.12 Memory Clusterer: Konu Bazlı Gruplama

Kümeleyici, dağınık hafıza dosyalarını anlamlı konu öbeklerine ayırır. İki tasarım kararı modülün davranışını belirler.

- Skor normalizasyonu doyum noktasına göre yapılır. Eşleşme sayısını konu sözlüğünün uzunluğuna bölen bir normalizasyon, zengin sözlüklü konuları sistematik olarak eler: 23 kelimelik bir “teknik” konusunda tek eşleşme minimum eşiğin altında kalırken, 9 kelimelik “aile” konusunda aynı tek eşleşme eşiği geçer. Doyum temelli normalizasyon bu yanlılığı ortadan kaldırır.
- Tetikleme açık bir bayrakla yapılır (--run). Yalnızca durum raporu basan bir çağrı (--stats), kümelemeyi hiç çalıştırmadığı halde çalışıyormuş izlenimi verir; iki işlev ayrı komutlarla ayrılmıştır.

3.13 Mood Journal: Ruh Hali Günlüğü

Veri kaynağı, retention metadata’sındaki duygu ve yoğunluk etiketleridir. İki kural hesabın geçerliliğini korur. Birincisi, yoğunluk ağırlıkları sıfırdan farklıdır: düşük yoğunluk “duygu yok” demek değildir, aksi halde düşük yoğunluklu bir öfke veya sevinç skora hiç katkı vermez. İkincisi, ruh hali yalnızca kullanıcı rolündeki kayıtlardan hesaplanır; asistan yanıtlarının sabit etiketlerle kaydedilmesi, mesajların yaklaşık yarısını yapay olarak nötr yapar ve skoru sistematik olarak sıfıra çeker.

3.14 Guidance Feedback: Davranışsal Geri Bildirim Döngüsü

Bu modül, Attention Gate’in ürettiği rehberliğin gerçekten işe yarayıp yaramadığını ölçer. Gecikmeli ödül mantığıyla çalışır: bir rehberliğin etkisi ancak bir sonraki mesajın duygusal valansı ile anlaşılabilir. Mimarideki gecikmeli ölçüm deseninin referans uygulamasıdır; espri kapanış defteri bu desenin ikinci örneğidir.

3.15 Memory Gist: Epizodik → Semantik Sıkıştırma

İnsan hafızasında ayrıntılı olay anıları zamanla soyut bilgiye dönüşür. Bu modül aynı dönüşümü uygular: tek tek konuşmalar, günler geçtikçe özetlenmiş semantik özlere indirgenir.

Epizodik : "20 Mayıs 14:32’de muhatap ‘Tokyo oteli pahalıymış’ dedi"
 ↓ (30+ gün sonra)
 Semantik : "Mayıs sonu: Tokyo oteli pahalı bulundu, alternatif aranıyor"

Bilinen bir sınırlılık: konu çıkarımı frekansı ikiden az olan kelimeleri eler. Bu, bir kez söylenen ama kritik bilgilerin (otel adı, doktor randevusu, verilen söz) özetle hiç iz bırakmaması anlamına gelir. Ham kayıt budandıktan sonra bu bilgi geri gelmez. Bkz. Bölüm 8.

3.16 Self-Reflection: Haftalık Öz-Değerlendirme

Haftalık öz-değerlendirme, beş ayrı modülün çıktısını birleştirerek sistemin kendi haftası hakkında yapılandırılmış bir rapor üretmesini sağlar: ruh hali, kişilik kayması, sosyal sinyaller (mizah performansı dahil), çıkarım isabeti ve watcher istatistikleri. Rapor, tüketicisi olan bir çıktıdır: ürettiği dersler iç anlatıya ve oradan uzun dönemli kimlik metnine akar.

Çalışma sıklığı ayrıca korumalıdır. Haftalık işi tetikleyen koşul “Pazar günü” olduğundan, Pazar’daki her heartbeat bu koşulu sağlıyor ve öz-değerlendirme aynı gün içinde tekrar tekrar çalışabiliyordu (13 Eylül 2026’da dört kez). Son çalışmadan bu yana en az altı gün geçmedikçe run() sessizce atlanır; --force bayrağı bu korumayı yok sayar.

3.17 Internal Narrative: Süren Benlik Anlatısı

NARRATIVE.md, ruh hali, etkin planlar, açık niyetler, merak kuyruğu ve son derslerden sentezlenen akıcı bir anlatıdır. Önemli ayrım: MEMORY.md olguları (ne olduğunu) tutarken, NARRATIVE.md hikayeyi (bunların sistem için ne anlama geldiğini) tutar. Bu, olgusal bellek ile otobiyografik benlik arasındaki ayrımın operasyonel karşılığıdır.

İç anlatı haftalık ölçekte tazelenir ve uzun dönemli kimlik işlevi görür; persona bloğu ise her mesajda güncellenen kısa ölçekli hattır. İkisi birbirinin yerine geçmez: anlatı süreklilik, blok ise anlık uyarlanma sağlar.

3.18 Active Learning: Zayıf Alan Tespiti

Aktif öğrenme katmanı, sistemin hangi konularda zayıf kaldığını izler ve gelişime en açık alanları önceliklendirir. Kayıt ve rapor tarafının aynı şema anahtarlarını kullanması bu modülde kritiktir; yazma ile okuma anahtarları arasındaki tek harflik bir uyumsuzluk dahi, başarı oranının aylarca sıfır raporlanmasına yol açabilir. Şema anahtarları bu nedenle tek bir sabitler tablosundan okunur.

3.19 Dreaming Loop: Hafıza Sentezi

Rüya döngüsü, insan uykusundaki bellek pekiştirme metaforunu sistemleştirir: yakın anıları sıkıştırır, beklenmedik bağlantılar arar ve örüntüleri soyutlar.

Bağlantı bulma fazının tümüyle gömme kalitesine bağlı olması, bu modülün en kritik tasarım kısıtıdır. Metin gömme fonksiyonu gerçek bir cümle kodlayıcıya (all-MiniLM-L6-v2, 384 boyut) bağlıdır; kodlayıcı yoksa karakter-trigram hash tabanlı bir yedek kullanılır. Hash ya da rastgele tohumlu vektörler semantik taşıyamaz: iki bağımsız Gaussian vektörün kosinüs benzerliği $\approx 0 \pm 0.05$ ’tir ve bu durumda 0.70 gibi bir eşik matematiksel olarak hiçbir zaman aşılamaz. Eşik 0.55’tir ve her anı benzersiz bir kaynak kimliği taşır; ortak kaynak etiketi taşıyan anılar birbirleriyle hiç karşılaştırılmaz.

Sıkıştırma oranı bir “içgörü” olarak değil, bir metrik olarak raporlanır. “5.7x sıkıştırma” bir bölme işleminin metne çevrilmiş halidir; içgörü listesine yazıldığında sistemin öğrenme çıktısı hakkında yanıltıcı bir bolluk izlenimi üretir.

3.20 Health Check: Sistem Sağlığı

Sağlık kontrolü heartbeat döngüsünün başında çalışır ve on iki metrik üretir. Dosya yaşına bakan yüzeysel bir tazelik kontrolü yerine, üç yapısal denetim yürütür:

- Hafıza bütünlüğü: dizi hizalaması, idx tekilliği, importance dağılım yüzdelikleri, yinelenen metin oranı, son gerçek anının tarihi.
- Sessiz çürüme: her state dosyası için modüle özgü bir bayatlama eşiği. Bir modülün hata vermeden yazmayı bırakması, hiçbir uyarı üretmeyen ve aylarca fark edilmeyebilen bir bozulma sınıfıdır; bu denetim o kör noktayı kapatır.
- Runtime köprüsü: bilişsel sistem prompt'a gerçekten bağlı mı? Eklenti yorumlayıcı yolu, otomatik geri getirme bayrağı, vektör indeksi büyüklüğü ve persona bloğunun tazeliği denetlenir. Bu zincirin kırılması sessizdir: yorumlayıcı yolu geçersizse hiçbir anı prompt'a girmez ve sistem yine de sorunsuz çalışıyor görünür.

Bu köprünün somut kanıtı yakın zamanda iki yönde görüldü. Enjeksiyon hattındaki persona ve arama çağrıları artık kalıcı bir worker sürecinde yürütülüyor; gömme modeli ve bilişsel durum yalnızca süreç başında bir kez yükleniyor. Ölçülen etki: arama 3.16 sn'den ~20 ms'ye, persona üretimi 3.18 sn'den ~15 ms'ye indi. Worker beş dakika içinde üç kez çökerse devre dışı kalır ve istekler eski tek seferlik yola düşer. Aynı denetim, rüya zamanlayıcısının 27 Mayıs 2026'dan beri fiilen donmuş olduğunu da ortaya çıkardı (bkz. Bölüm 8.2).

3.21 Canlı Hattın Dışındaki Modüller

Dokuz modül arşivde tutulur. Bu modüllerin fikirleri değerliydi; uygulamaları çalışmıyordu. Ana dizinde durdukları sürece modül sayımını şişiriyor, hem sağlık raporlarını hem de mimari tarifini kirletiyorlardı. Silinmek yerine arşivlenmeleri bilinçli bir tercihtir: her birinin neden arşivlendiği ve canlıya nasıl döndürülebileceği belgelenmiştir.

MODÜL	ARŞİV GEREKÇESİ
sensory_interface	391 satır, sıfır içe aktarım. Durum dosyası hiç oluşmadı; sınıf bir kez bile örneklenmedi.
curiosity_engine	Soru bankası ve keşif günlüğü boş; ilgi alanları yapısal kelimelerle doldu. Yerini surprise_curiosity aldı.
predictive_coding	Olay temsili ham 50 karakterlik metin parçasıydı; her olay tekil olduğu için her şey maksimum sürpriz üretiyordu.
self_supervised_loop	Öğrenme adımı literal random.random() idi ve sahte sonuçlar kalıcı takvime yazılıyordu.
meta_learner	Ürettiği hiperparametreleri hiçbir modül okumuyordu. Tek noktalı sonlu fark gradyanı, gürültülü ölçümle anlamsız.
self_evolution	Fitness ağırlıkları ölçülen metriklerle hizasızdı; kesişim tek anahtardı, yani fitness = adaptability.
world_model	Halefiyle işlev çakışması. Tek "yaratıcı içgörü"nün tüm ağırlıkları başlangıç değerinde kalmıştı.

MODÜL	ARŞİV GEREKÇESİ
cognitive_trainer	Eğitim, semantik taşımayan gömmeler üzerine yapılmıştı; öğrenilen şey gürültünün ezberiydi.
cognitive_net	61.2M parametrelili mimari tutarlı; sorun eğitim verisindeydi. Hiçbir canlı yol modeli yüklemiyordu.

Canlıya geri alma ölçütleri

1. Modülü besleyen gerçek bir veri yolu olmalı; sahte veya tautolojik metrik üretmemeli.
2. Çıktısını tüketen bir yer belirlenmeli: duruma yazmak yetmez, davranışı etkilemeli.
3. Durum dosyası, sağlık denetimindeki bayatlama listesine eklenmeli.

4. Veri Yapıları ve State Yönetimi

4.1 State dosyaları haritası

Sistemin tüm belleği diskte yaşar. Aşağıdaki harita ana hafıza dosyalarını ve modül-spesifik durum dosyalarını gösterir.

```
workspace/
├─ MEMORY.md                curated uzun dönem hafıza
├─ NARRATIVE.md             süren benlik anlatısı
├─ SOUL.md · USER.md       kimlik ve kullanıcı tanımı
├─ HEARTBEAT.md · AGENTS.md çalışma talimatları
├─ retention_state.json     konuşma anıları + metadata
├─ memory/YYYY-MM-DD.md    günlük ham notlar
├─ cognitive_state/
│   ├─ persona_block.md     prompt'a yazılan blok
│   ├─ affect_state.json    valence / arousal durumu
│   ├─ personality_development.json
│   ├─ inference_state.json
│   ├─ metacognition.json
│   ├─ social_intelligence.json (espri defteri dahil)
│   ├─ guidance_feedback.json
│   ├─ curiosity_queue.json
│   ├─ prospective_memory.json
│   ├─ semantic_gists.json
│   ├─ memory_clusters.json
│   ├─ self_reflection.json
│   ├─ internal_narrative.json
│   ├─ watcher_log.json
│   └─ active_learning.json
```

4.2 Atomik IO

Tüm durum yazımları geçici dosya → fsync → atomik yer değiştirme sırasıyla yapılır. Bozuk dosya okunduğunda varsayılan değer döner ve sistem çökmez.

Retention kaydetme yolu tek geçişlidir. Aynı dosyaya iki ayrı yazım yapan bir tasarım (önce düz bir json.dump, sonra dosyayı tekrar okuyup metadata ekleyen ikinci bir yazım) iki maliyet üretir: iki yazım arasında yaşanacak bir çökmede tüm importance ve duygu metadata'sı kalıcı olarak kaybolur, ayrıca her mesajda megabayt ölçeğindeki dosya iki kez diske yazılır. Tek geçişli yazım her iki riski de ortadan kaldırır.

4.3 Retention state yapısı

```
{
  "version": "4.0",
  "conversation_memory": [ "2026-07-11 11:07: [user] ...", ... ],
  "memory_metadata": [
    { "idx": 0, "emotion": "happy", "intensity": "medium",
      "importance": 0.63, "role": "user",
      "timestamp": "2026-07-11 11:06",
      "text_preview": "...", "stored_at": "...",
      "review_count": 0, "decay_state": "full" }, ...
  ],
  "last_decay": "2026-07-29T15:43:09",
  "personality_state": [ ... ]
}
```

5. Algoritmalar ve Matematiksel Temeller

Bu bölüm, mimarının sayısal çekirdeğini oluşturan formülleri ve her birinin tasarım gerekçesini verir. Yeniden üretim (replication) yapacak okuyucu için kritik referanstır.

5.1 Önem skoru (importance)

Önem skoru, bir anının ne kadar süre ve ne öncelikle saklanacağını belirleyen merkezi büyüklüktür. Model toplamsal ağırlıklıdır:

```
importance = 0.35·w_emotion + 0.25·w_intensity  
            + 0.25·w_topic   + 0.15·w_repetition  
w_topic     = min( $\sum$  topic_boost / 0.6, 1)  
w_repetition = min(eşleşen_konu_sayacı / 8, 1)  
her bileşen [0,1] normalize; sonuç doğal olarak aralıkta kalır.
```

Toplamsal modelin tercih edilme gerekçesi ölçek davranışıdır. Çarpımsal bir model (intensity × emotion × katsayı + konu + tekrar), nötr olmayan bir duygu ile yüksek yoğunluk bir araya geldiğinde tek başına tavanı deler; konu ve tekrar bileşenleri kırpma sırasında tamamen yutulur. Sonuç, iki değerli (bimodal) bir dağılımdır: skor ya taban ya tavan değerini alır. Böyle bir skor ayırım gücü taşımaz; oysa filtreleme, decay kovası seçimi, aralıklı tekrar eşiği ve özet seçimi hep bu skora dayanır. Toplamsal modelin canlı veri üzerindeki ölçülen dağılımı gerçek bir yayılım gösterir: ortanca 0.30, doksanıncı yüzdelik 0.785.

Asistan (rol=assistant) kayıtlarında bu sonuç ayrıca ×0.7 ile ölçeklenir; aksi halde duygu ve emoji yüklü asistan cevapları kullanıcının kısa mesajlarını önem sıralamasında sistematik olarak geride bırakıyordu. Bkz. Bölüm 3.2.

5.2 Hibrit geri getirme (RRF)

İki aday listesi Reciprocal Rank Fusion ile birleştirilir, ardından önem ve tazeliikle yeniden sıralanır.

```
dense      = TF-IDF (char_wb 2-5 gram) kosinüs sıralaması, ilk 20  
lexical    = BM25-lite (IDF ağırlıklı, uzunluk normalize) sıralaması, ilk 20  
rrf(d)     =  $\sum 1 / (60 + \text{rank}_i(d))$            $i \in \{\text{dense}, \text{lexical}\}$   
final(d)   = 0.60 · rrf_norm(d)  
            + 0.25 · importance(d)  
            + 0.15 · 0.5^(yaş_gün / 30)
```

Uygulamada kritik nokta, vektörleyicinin ne zaman fit edildiğidir. Vektörleyici ilk encode() çağrısında fit ediliyorsa ve arama fonksiyonu önce sorguyu encode ediyorsa, sözlük tamamen sorgunun karakter n-gramlarından oluşur, IDF tek dokümana göre hesaplanır ve sorguda geçmeyen n-gram taşıyan anılar sıfır vektöre düşer. Bu durumda “semantik arama” pratikte bulanık bir substring eşleşmesine indirgenir. Vektörleyici bu nedenle korpus üzerinde fit edilir ve süreç ömrü boyunca sorguya kilitlenmez.

5.3 Affect dinamiği

```
(v, a) = to_va(emotion, intensity)
emotion_v ← emotion_v + 0.50 · (v - emotion_v)    hızlı
mood_v    ← mood_v    + 0.05 · (v - mood_v)       yavaş
Sessizlikte zaman gerilemesi:
  mood_v    ← mood_v    · 0.5^(Δt_saat / 36)
  emotion_v ← emotion_v · 0.5^(Δt_saat / 1)
Alışma (habituation):
  delta ← delta · 1 / (1 + 0.3 · son_5te_ayni_durum_sayisi)
```

Yoğunluk ağırlıkları tek bir tabloda toplanmıştır: low 0.5, medium 1.0, high 1.5, extreme 2.0. Düşük yoğunluğun sıfırla çarpılması psikolojik olarak yanlıştır: düşük yoğunluk “duygu yok” anlamına gelmez, yalnızca daha zayıf bir sinyal taşır.

5.4 Trait güncelleme matematiği

```
λ_t = decay_rate_t / 0.003 × 0.02
trait ← trait + (baseline_t - trait) · λ_t    setpoint
trait ← trait + habituation · (Δ_durum + Δ_konu) öğrenme
trait ← clamp(trait, 0.05, 1.0)
Sessizlikte (etkileşim yokken):
  pull = 1 - 0.5^(Δt_gün / 21)
  trait ← trait + (baseline_t - trait) · pull    (üst sınır 90 gün)
```

Setpoint modeli iki problemi aynı anda çözer. Mutlak azalma ($\text{trait} \times \text{decay_rate} \times \Delta t$) kullanan bir model, öğrenme adımı decay adımından büyük olduğunda dengeyi aralığın dışına taşır ve trait tavana yapışır; ters uçta ise nadiren pekiştirilen bir trait tabana çöker ve kimlik tanımıyla çelişir. Çekim merkezine dönüş her iki uçta da dengeyi aralığın içinde tutar ve “kişilik değişir ama özü kalır” semantiğini modele doğrudan sokar.

Konu modülasyonu ayrıca ele alınmalıdır: bu deltalar ana kırpmadan sonra eklenirse normal öğrenme adımının katları büyüklüğe ulaşabilir ve tek bir kelime eşleşmesi bir trait’i tek adımda sıçratabilir. Bu nedenle konu deltaları ayrı ve daha küçük bir tavana (± 0.012) tabidir.

5.5 Mizah kalibrasyonu: Beta-Bernoulli

Ham oran ($1/1 = \%100$) yerine her hücre için bir Beta dağılımı tutulur. Hücreler konu $\times$ zaman dilimi $\times$ alt tip kırılımındadır.

```
landed → α ← α + max(laugh_strength, 0.3)
bombed → β ← β + 1.0
ignored → β ← β + 0.4                      (zayıf negatif kanıt)
posterior = Beta(α, β)
mean      = α / (α + β)
n         = α + β - 2
seviye = 0.5 + (mean - 0.5) · min(n/10, 1) belirsizlikte ortaya çek
karar   = seviye ≥ 0.45 ve cooldown = 0
```

Beta modeli az veriyle otomatik bir keşif-sömürü dengesi kurar ve tek gözlemden kesin sonuç çıkarma problemini matematiksel olarak çözer. Depolama maliyeti hücre başına iki kayan noktalı sayıdır.

5.6 Güven kalibrasyonu

```
hit_rate = (confirmed + 1) / (confirmed + refuted + 2)    Laplace
raw       = hit_rate / avg_conf
shrink    = min(çözülen / 10, 1)
factor    = clamp(1 + (raw - 1) * shrink, 0.5, 1.2)
brier     = mean( (conf - outcome)2 )    0 = mükemmel, 0.25 = rastgele
kovalar   : beyan edilen güven → gözlenen isabet (0.1 aralıklarla)
```

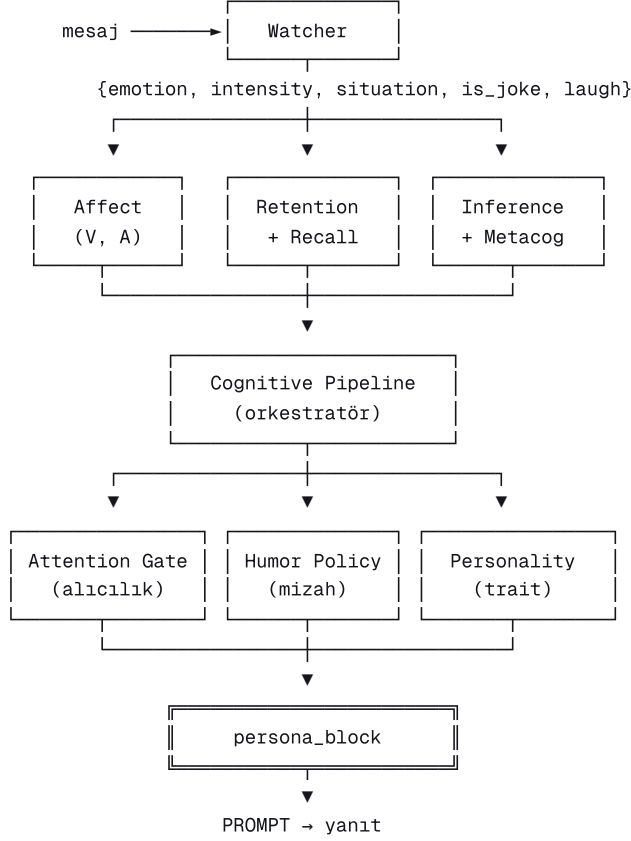
5.7 Hafıza zayıflaması

```
yaş = şimdi - kaydedilme_zamanı
kova = importance_kovası(importance)
yaş > kova.sil      → kayıt düşer
yaş > kova.arşivle  → metadata.decay_state = "archived"
yaş > kova.tam_tut  → metadata.decay_state = "faded"
aksi halde         → "full"
Metin hiçbir durumda değiştirilmez; idx eşlemesi yeniden kurulur.
```

6. Modüller Arası Veri Akışı

6.1 Ana veri akış şeması

Watcher'dan çıkan sinyal üç ana yöne dağılır; tüm sonuçlar persona bloğunda toplanıp yanıt üretimine geri döner. Şemadaki çift çizgili yol, mimariyi kapalı döngü yapan geri dönüş hattıdır.



6.2 Modül bağımlılık özeti

MODÜL	GİRDİ ALDIĞI	ÇIKTI VERDİĞİ
Watcher	ham mesaj	Affect, Retention, Inference, Personality, Social, Humor
Affect	Watcher	Humor Policy, persona_block
Retention	Watcher	Recall, Mood Journal, Gist, Dreaming, persona_block
Inference	Watcher, Retention	Metacognition, proaktif sorular

MODÜL	GİRDİ ALDIĞI	ÇIKTI VERDİĞİ
Attention Gate	Watcher log	Guidance Feedback, Humor Policy, persona_block
Humor Policy	Gate, Social, Personality, Affect	persona_block
Social Trainer	Watcher (espri + tepki)	Humor Policy, Self-Reflection
Personality	Watcher, Affect, importance	persona_block, Self-Reflection
Self-Reflection	Mood, Personality, Social, Metacog	Internal Narrative, NARRATIVE.md

6.3 Kritik veri yolları

- Duygu → hafıza → ruh hali → anlatı: bir mesajın duygusal etiketi önem skorunu, o da saklama süresini ve özet seçimini belirler; haftalık anlatı bu birikimden beslenir.
- Espri → tepki → posterior → politika → prompt: bir esprinin sonucu bir sonraki mesajda ölçülür ve gelecekteki mizah kararını değiştirir.
- Rehberlik → sonraki valans → istatistik: Attention Gate'in tavsiyesi bir sonraki mesajın duygusal tonuyla değerlendirilir.
- Çıkarım → tahmin → doğrulama → kalibrasyon: her çıkarım açık bir tahmin olarak kaydedilir ve sonradan gerçekleşip gerçekleşmediğine göre gelecekteki güveni ayarlar.

7. Doğrulama ve Ölçüm Dürüstlüğü

Kendi durumunu kendi ürettiği metriklerle raporlayan bir sistem yapısal olarak iyimserdir: metriklerin bozulması, raporun bozulmasıyla eşzamanlıdır, yani bozulma kendini gizler. Bu nedenle doğrulama, mimarının dışında bir bakım faaliyeti değil, mimarının bir parçası olarak kurgulanmıştır. Yöntemsel katkı iddiası buradadır: kendi kendini raporlayan bilişsel bir sistemin nasıl denetleneceği, literatürde az işlenmiş bir konudur.

7.1 Regresyon testleri

Düzeltilen her hata için bir test yazılır. Testlerin belgeleme dizesi hangi hatayı koruduğunu anlatır; bir test kırılırsa o hata geri gelmiş demektir. Test kümesi halihazırda 64 testten oluşur ve modül içe aktarımından uçtan uca persona bloğu üretimine kadar uzanır.

```
def test_dream_embedding_is_semantic():
    """embed_text saf gürültü olduğunda (tohumlanmış randn) döngüde
    hiç bağlantı bulunmaması matematiksel zorunluluktur."""
    a = dl.embed_text("kahve içmeyi çok seviyorum")
    b = dl.embed_text("kahve içmek güzel bir alışkanlık")
    c = dl.embed_text("uçak bileti fiyatları arttı")
    assert cos(a, b) > cos(a, c)
```

7.2 Sağlık denetimi

Heartbeat döngüsünün başında on iki metrikli bir sağlık kontrolü çalışır. Üç denetim, kendi kendini raporlayan sistemlerin klasik kör noktalarını hedefler: hafıza bütünlüğü (dizi hizalaması, kimlik tekilliği, önem dağılımı, tekrar oranı), sessiz çürüme (modül başına bayatlama eşiği) ve runtime köprüsü (bilişsel sistem prompt'a gerçekten bağlı mı). Ayrıntı için bkz. Bölüm 3.20.

7.3 Göç betikleri

Veri temizliği ve şema birleştirme işlemleri tek seferlik betikler olarak kodlanır; her biri önce kuru çalıştırma (dry-run) yapar, uygulamadan önce yedek alır ve ne yaptığını raporlar. Bu, hem tekrarlanabilirlik hem de geri alınabilirlik sağlar.

7.4 İzlenen hata sınıfları

Doğrulama rejimi, aşağıdaki hata sınıflarını açıkça hedefler. Sınıflandırmanın kendisi bir tasarım aracıdır: yeni bir modül yazılırken bu liste bir kontrol listesi olarak kullanılır.

SINIF	TANIM	TİPİK BELİRTİ
Kopuk zincir	Üretilen çıktının tüketicisi yok	Trait vektörü hesaplanıyor, prompt'a girmiyor

SINIF	TANIM	TİPİK BELİRTİ
Sessiz başarısızlık	Hata yakalanıp yutuluyor	Geçersiz yorumlayıcı yolu → geri getirme ölü
Erişilemez kod	Mantıksal olarak ulaşılamayan dal	Üstteki koşulun zaten kapsadığı bir elif
Ölçek hatası	Formül tasarlandığı aralığın dışına taşıyor	Çarpımsal skorun tavanı delmesi
Şema kayması	Aynı dosyaya farklı şemalarla yazılıyor	Üç modül, üç farklı kişilik şeması
Sahte metrik	Rastgele veya tautolojik değer	“Öğrenme” adımının literal <code>random.random()</code> olması
Örneklem yanılığı	Tek gözlemden kesin sonuç	“Espri başarısı %100” (n = 1)
Dilsel kırılğanlık	Türkçe’ye özgü ayrıştırma hatası	Tipografik kesme, kelime sınırı, ek çekimi

Son sınıf özellikle dikkate değerdir. Türkçe eklemeli bir dildir ve kelime sınırı temelli desen eşleştirme, İngilizce için tasarlanmış varsayımlarla çalışmaz. “usa” ekinin Asakusa içinde eşleşmesi, “tr” kısaltmasının trafik içinde eşleşmesi, “cuma” gününün cumartesiye yutması: bunların hepsi aynı kök nedene sahiptir. Mimaride tek taraflı sınır kuralı benimsenmiştir: sol sınır zorunlu, sağ sınır serbest (kelime ek alabilirsin), üç harften kısa tokenlarda iki taraflı sınır zorunlu.

7.5 Doğrulama rejiminin göremedikleri

Yöntem statik ve durum-temellidir; şu üç şeyi göremez. Birincisi, doğru çalışan ama yanlış tasarlanmış bir mekanizmayı; kod niyetine uygun çalışıyorsa denetim onu onaylar. İkincisi, henüz veri birikmemiş bir modülün gelecekteki davranışını. Üçüncüsü, dil modelinin persona bloğunu gerçekten dikkate alıp almadığını; bu ancak kontrollü bir A/B ölçümüyle doğrulanabilir ve Bölüm 8’de açık bir çalışma başlığı olarak listelenmiştir.

8. Sınırlılıklar ve Gelecek Çalışma

Bu bölüm, sistemin bilinen zayıflıklarını açıkça listeler. Bir mimarının akademik değeri, neyi başardığı kadar neyi başaramadığını da bilmesinden gelir.

8.1 Doğrulanmamış iddialar

- Persona bloğunun etkisi ölçülmemiştir. Blok prompt'a giriyor, ancak dil modelinin yanıtını ölçülebilir biçimde değiştirip değiştirmediği kontrollü bir karşılaştırmayla test edilmedi. Önerilen yöntem: blok enjekte edilen ve edilmeyen yanıtları izleme katmanından geçirip negatif sinyal oranını ve kullanıcı tepkisini karşılaştırmak.
- Mizah kalibrasyonunun örnekleme henüz çok küçüktür. Beta posterior'ları matematiksel olarak doğru çalışsa da anlamlı bir kalibrasyon için en az yirmi kapanmış espri gerekir.
- Kullanıcı sayısı birdir. Sistem tek bir muhatapla geliştirilmiştir; çoklu profil mimarisi tasarlanmış ancak gerçek çoklu-kullanıcı verisiyle test edilmemiştir.

8.2 Yapısal sınırlar

- Çıkarım motoru deterministiktir. Elle yazılmış Türkçe kalıplara dayanır; kalıp dışına çıkan ifadeleri göremez. Önerilen yön: kalıplar hızlı yol olarak kalsın, eşleşme yoksa ve mesajda ilgili sinyal varsa tek bir yapılandırılmış dil modeli çağrısıyla varlık/olgu/plan çıkarımı yapılsın.
- Rüya döngüsü heartbeat yolunda fiilen donmuş durumda. Zamanlayıcı son gerçek döngüsünü 27 Mayıs 2026'da tamamladı (toplam 6 döngü) ve bir sonraki çalışma damgası hala 11 Mayıs'ta duruyor; heartbeat ortamında gömme modelinin bağımlılığı yüklenemediğinde tetiklenen her çalışma sessizce çöküyor. Sağlık kontrolü bunu artık yeşil yerine uyarı olarak doğru raporluyor; döngünün kendisi henüz onarılmadı.
- Özetleme tek geçen bilgiyi eler. Semantik sıkıştırma, frekansı ikiden az olan kelimeleri konu listesine almaz; bir kez söylenen bir otel adı veya verilen bir söz özette iz bırakmaz. Ham kayıt budandıktan sonra bu bilgi geri gelmez.
- Duygu çözünürlüğü yedi etiketle sınırlıdır. Gerçek veride pratikte dört etiket görülür. Türkçe için yüksek getirili adaylar: hayal kırıklığı, minnettarlık, gurur.
- Kişilik sabit bir taksonomiye dayanır. On yedi trait elle seçilmiştir; veriden türetilmiş bir boyut analizi yapılmamıştır.

8.3 Öncelikli gelecek çalışma

BAŞLIK	NEDEN	BEKLENEN ETKİ
Persona bloğu A/B ölçümü	Mimarının merkezi iddiası doğrulanmamış	Temel katkının kanıtı

BAŞLIK	NEDEN	BEKLENEN ETKİ
Hibrit çıkarım (kalıp + LLM)	Kalıplar gerçek veride düşük isabet veriyor	Plan ve niyet yakalamada sıkırama
Kullanım-temelli önem	Elle yazılmış anahtar kelime tablosu ölçeklenmiyor	İnsan hafızasına daha yakın model
Duygu paletinin genişletilmesi	Dört etiketle duygusal derinlik sınırlı	Daha ince ruh hali takibi
Çoklu kullanıcı doğrulaması	Profil mimarisi test edilmemiş	Genellenebilirlik iddiası

8.4 Kapanış değerlendirme

Hybrid Cognitive System, insan bilişinden ilham alan bir asistan mimarisinin tek kullanıcılı, uzun süreli ve kesintisiz bir uygulamasıdır. Güçlü yanı kavramsal tutarlılığıdır: duygu → önem → unutma → özetleme zinciri, bağlamsal trait aktivasyonu ve alıcılık temelli sosyal zamanlama, ilgili literatürle doğrudan ilişkilendirilebilir tasarım kararlarıdır.

Mimarinin merkezi iddiası, bu kavramsal zenginliğin davranışa bağlanmış olmasıdır: enjeksiyon katmanı, ölçülen her sinyal için bir tüketici tanımlar. Ancak bağın kurulmuş olması, etkisinin ölçülmüş olduğu anlamına gelmez; bu ayrımı korumak, mimarinin kendi doğrulama rejiminden çıkardığı en önemli derstir.

Metnin ikinci katkısı yöntemeldir. Kendi kendini raporlayan bir sistemin nasıl denetleneceği, hangi hata sınıflarının sessiz kaldığı ve ölçüm dürüstlüğü'nün nasıl bir tasarım özelliğine dönüştürüleceği: bunlar tek bir sisteme özgü sorular değildir. Bölüm 7, bu soruların somut bir mimaride nasıl ele alındığını belgeler.

Ek A: Komut Referansı

Sistemin canlı durumunu incelemek ve bakımını yapmak için kullanılan komutlar. Tümü hybrid_cognitive dizininden çalıştırılır.

```
# Konuşma sonu: TEK komut (kişilik güncellemesi dahil)
python3 cognitive_pipeline.py --pair "<mesaj>" "<cevap>"

# Durum ve tanı
python3 cognitive_pipeline.py --status --json      # tam durum + sağlık
python3 cognitive_pipeline.py --modules --json     # modül registry
python3 cognitive_pipeline.py --persona           # persona bloğu
python3 health_check.py --json                   # 12 metrikli sağlık

# Hafıza
python3 cognitive_retention.py --health --json     # bütünlük denetimi
python3 cognitive_retention.py --search "kelime"   # hibrit arama
python3 cognitive_retention.py --recall "konu"     # recall bloğu
python3 cognitive_retention.py --decay-check      # yaşlandırma

# Duygu, kişilik, mizah
python3 affect.py --show                         # valence/arousal
python3 mood_journal.py --week                   # ruh hali
python3 update_personality.py --profile           # trait profili
python3 humor_policy.py "son mesaj"              # mizah kararı
python3 social_trainer.py                       # espri istatistikleri

# Bakım
python3 memory_clusterer.py --run                 # kümeleme
python3 dreaming_loop.py --cycle 1               # rüya döngüsü
python3 memory_gist.py --run                     # semantik sıkıştırma
python3 self_reflection.py --run                 # haftalık değerlendirme
python3 prospective_memory.py --check            # niyet kontrolü
python3 prospective_memory.py --ack <id>         # teslim onayı

# Doğrulama
python3 tests/run.py                             # 64 regresyon testi
```

Ek B: Terminoloji

TERİM	KARŞILIĞI
persona bloğu	Yanıt üretiminden önce prompt'a yazılan, sistemin anlık bilişsel durumunu özetleyen metin
alıcılık (receptivity)	Muhatabın şu an etkileşime ne kadar açık olduğunun 0–1 arası nicel tahmini
mizaç temeli (setpoint)	Bir trait'in uzun dönemde döndüğü denge değeri; profil tipinin başlangıç değeri
alışma (habituation)	Tekrarlanan aynı uyarının etkisinin kademeli olarak azalması
kapanış defteri	Bir müdahalenin (espri, ton tavsiyesi) sonucunun bir sonraki mesajda ölçülmesi için tutulan bekleyen kayıt listesi
geri dönen espri (callback)	Daha önce tutmuş bir espriye sonradan yapılan gönderme
sessiz çürüme	Bir modülün hata vermeden çalışmayı bırakması ve bunun fark edilmemesi
ölçüm dürüstlüğü	Her istatistiğin örneklem büyüklüğü ve güven düzeyiyle birlikte raporlanması
epizodik → semantik	Ayrıntılı olay anılarının zamanla soyut bilgiye dönüşmesi
RRF	Reciprocal Rank Fusion: farklı arama yöntemlerinin sıralamalarını birleştirme tekniği

Bu belge sistemin ölçülmüş durumunu yansıtır. Sayısal değerler canlı durum dosyalarından okunmuştur; ölçülmemiş iddialar Bölüm 8'de sınırlılık olarak listelenmiştir.

ATIF

Yaşlıoğlu, M. (2026). Hybrid Cognitive System: Durum-bilinçli, kişilik geliştiren ve kapalı döngülü bilişsel asistan mimarisi [Çalışma makalesi].
<https://yaslioglu.com/tr/projeler/skynet/makale/>