Dış servis düştüğünde

Timeout, retry ve circuit breaker: yavaş bir dış servisin tüm uygulamayı kilitlemesini üç savunma hattıyla nasıl durdurursun.

01 ÇÖKMEDİ, YAVAŞLADI

Bir SMS sağlayıcısı yavaşladı. Bütün uygulama durdu.

Yanıt süresi 30 saniyeye çıktı; yarım saat içinde SMS ile hiç ilgisi olmayan endpoint'ler bile zaman aşımına uğruyordu.

02 CASCADING FAILURE

Yavaşlık nasıl yayılır

1. **Process kilitlenir** — `worker / PHP-FPM process bağlı`
2. **Süre kaybolur** — `30 sn çağrı = 30 sn meşgul`
3. **Havuz dolar** — `yeni istekler sırada bekler`
4. **Sağlıklılar ölür** — `ilgisiz endpoint'ler timeout`

5 sn

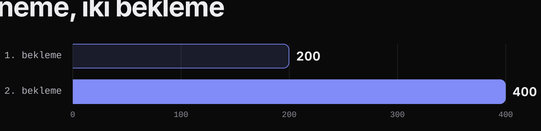
03 BİRİNCİ SAVUNMA: TIMEOUT

toplam timeout

Varsayılan 30 saniye değil, 5: Http::timeout(5)->connectTimeout(2). Değerler kaynağın önerdiği örnektir; bu sayı bir karar olmalı.

04 İKİNCİ SAVUNMA: SINIRLI RETRY

Üç deneme, iki bekleme



Örnek değerler: 200 × 2^{n-1} ms, üstüne random_int(0, 100) ms jitter. Üç deneme arasında iki bekleme olur; hâlâ hata varsa sorun geçici değildir.

~3,4 dk

05 BACKOFF'A TAVAN KOY

cap olmadan 12. bekleme

delay = min(cap, base × $factor^{n-1}$). Cap olmadan base 100 ms × 2, 12. denemede yaklaşık 3,4 dakikaya ulaşır.

06 THUNDERING HERD

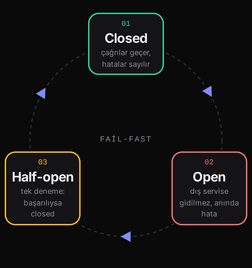
Jitter türleri

TÜR	ARALIK	DAVRANIŞ
Full	`[0, delay]`	En geniş dağılım; bir istemci çok erken deneyebilir
Equal	`[delay/2, delay]`	Bir taban tutar, yine de herd'i dağıtır
Decorrelated	`[base, önceki × 3]`	Capped; son beklemeye uyar, güçlü varsayılan
None	`jitter yok`	Tek istemci için sorun değil, bir filo için tehlikeli

07 ÜÇÜNCÜ SAVUNMA: CIRCUIT BREAKER

Çağrı yapmayı bırakmak

5 saniyelik timeout bile process başına 5 saniyedir; open durumda bu süre 0'a iner.



08 FAİL-FAST, YAVAŞ-FAİL'DEN İYİDİR

Devre açıkken ne yaparsın

Kuyruğa al
SMS hemen gitmek zorunda değilse devre kapanınca işlenmek üzere bırak.

Bayat veriyi sun
Döviz kuru servisi çöktüyse son bilinen kuru göster.

Zarifçe kıs
"Öneriler şu an yüklenemiyor" demek, sayfayı çökertmekten iyidir.