Solicitação de apoio técnico à RTM - Instabilidade no serviço HSM (respostas 504 em 3 segundos)

- De: Equipe técnica Monetarie SCD (ISPB 46026562)
- Para: Time técnico RTM
- Data: 24/07/2026
- Ambientes: vHSM 60042; HML `https://monetarie-hsm-hml.priv.rtmcloud.net.br`; PRD `https://monetarie-hsm-prd.priv.rtmcloud.net.br`
- PDF entregável: `docs/reports/2026-07-24-hsm-rtm-solicitacao-apoio-504.pdf` (gerado deste conteúdo)

1. Resumo

Estamos observando, nos dois ambientes (homologação e produção), uma fração alta de respostas HTTP 504 do gateway do HSM (EcosCryptoServer) com corte consistente em 3 segundos. O corpo da resposta indica um timeout interno de 3 segundos entre o gateway e o vHSM. O HSM está no caminho crítico de toda assinatura e decifra de mensagens que trocamos com o Banco Central (PIX e SPB). O nosso sistema foi projetado e ajustado para operar abaixo de 500 ms por transação, e cada corte de 3 segundos do HSM rompe diretamente essa margem de engenharia e nos impacta. Precisamos do apoio da RTM para validar essas ocorrências nos logs internos, nos orientar sobre a causa e ajustar o que for necessário do lado do serviço, porque estamos em rampa de aumento de volume transacional.

Já fizemos a nossa parte da lição de casa: inventariamos todas as chamadas que nossa aplicação faz ao HSM, reduzimos em mais de 90% o volume que gerávamos (era dominado por handshakes mTLS) e implementamos retry com backoff. Mesmo com o nosso volume reduzido, os 504 continuam ocorrendo, inclusive em chamadas de health que não tocam o vHSM, o que aponta para degradação do lado do serviço e não para sobrecarga nossa.

2. Contexto: por que isso nos afeta tanto

Por desenho de segurança, a chave privada da Monetarie nunca sai do HSM: toda assinatura (mensagens de saída ao BACEN) e toda decifra (mensagens de entrada) passam pelo vHSM 60042. Cada 504 de 3 segundos entra direto na latência de uma transação financeira ou derruba o processamento de uma mensagem recebida. Como o nosso sistema é trabalhado para operar abaixo de 500 ms por transação, uma taxa de falha de 50 a 67% por chamada, com custo de 3 segundos por falha, é incompatível com a nossa operação.

3. O que estamos observando

3.1 Medição controlada por endpoint (23/07/2026)

Probe com chamadas sequenciais espaçadas de 700 ms (sem concorrência), timeout do cliente em 8000 ms, medindo status, latência e corpo.

HML (6 amostras por endpoint):

Endpoint	OK	Erro	Latências OK (ms)	Latências erro (ms)	Status
GET /v1/health	6/6	0	37, 3043, 3045, 3044, 22, 22	-	-

Endpoint	OK	Erro	Latências OK (ms)	Latências erro (ms)	Status
POST get-session-credential	3/6	3	240, 251, 239	3007, 3010, 3021	504
POST sign-rsa	2/6	4	64, 27	3007, 3018, 3026, 3019	504

PRD (3 amostras por endpoint):

Endpoint	OK	Erro	Latências OK (ms)	Latências erro (ms)	Status
GET /v1/health	3/3	0	3182, 3057, 3060	-	-
POST get-session-credential	2/3	1	745, 652	3005	504
POST sign-rsa	1/3	2	33	3004, 3024	504

Corpo do 504, idêntico em todas as ocorrências nos dois ambientes:

```
{"returnValue":"The request was canceled due to the configured HttpClient.Timeout of 3 seconds elapsing."}
```

Três leituras diretas dessa medição:

1. O corte acontece sempre em torno de 3000 ms (3004 a 3026 ms) e o corpo cita o HttpClient.Timeout de 3 segundos. O nosso cliente espera 8 segundos, portanto quem encerra a chamada é o gateway.
2. O GET /v1/health, que não abre sessão nem executa operação criptográfica, respondeu 200 mas levou cerca de 3 segundos em várias amostras (produção: nas 3 de 3). Lentidão nesse endpoint independe do nosso volume de criptografia.
3. O padrão bimodal (respostas em dezenas de milissegundos ou corte em 3 segundos) é típico de backend lento ou saturado atrás de um proxy com timeout fixo.

3.2 Ocorrências contínuas em produção (noite de 23/07 para 24/07, horários UTC)

No restart da nossa aplicação às 02:03 UTC, o autoteste de conectividade registrou 504 em sequência tanto em get-session-credential quanto em sign-rsa, para os cinco destinos testados, normalizando apenas após novas tentativas:

```
2026/07/24 02:03:03 [selftest] dict     HANDSHAKE FAILED: get-session-credential HTTP 504
2026/07/24 02:03:06 [selftest] dict_np  HANDSHAKE FAILED: get-session-credential HTTP 504
2026/07/24 02:03:10 [selftest] icom     HANDSHAKE FAILED: get-session-credential HTTP 504
2026/07/24 02:03:13 [selftest] icom_sec HANDSHAKE FAILED: get-session-credential HTTP 504
2026/07/24 02:03:16 [selftest] arq      HANDSHAKE FAILED: get-session-credential HTTP 504
2026/07/24 02:03:29 [selftest] dict     HANDSHAKE FAILED: sign-rsa HTTP 504
2026/07/24 02:03:33 [selftest] icom_sec HANDSHAKE FAILED: sign-rsa HTTP 504
2026/07/24 02:06:08 [selftest] icom_sec HANDSHAKE FAILED: sign-rsa HTTP 504
```

Entre 02:10 e 03:07 UTC, nosso worker de echo periódico (uma mensagem assinada a cada 20 segundos) registrou 41 falhas de assinatura por 504 em cerca de 171 tentativas. Importante: cada uma dessas 41 falhas já é o resultado FINAL após 3 tentativas com backoff, ou seja, cerca de 24% das operações falharam mesmo com retry. Essa taxa final é matematicamente consistente com a taxa por chamada de 50 a 67% medida no probe (0,62 ao cubo é aproximadamente 0,24). Amostra:

```
02:35:45.845 [warning] echo pibr.001 falhou canal=primary reason={:signing_failed, "RTM HSM
signing failed: {:http_status, 504}"}
02:38:50.181 [warning] echo pibr.001 falhou canal=primary reason={:signing_failed, "RTM HSM
signing failed: {:http_status, 504}"}
02:40:23.138 [warning] echo pibr.001 falhou canal=primary reason={:signing_failed, "RTM HSM
signing failed: {:http_status, 504}"}
02:43:46.627 [warning] echo pibr.001 falhou canal=primary reason={:signing_failed, "RTM HSM
signing failed: {:http_status, 504}"}
```

3.3 Caso concreto de impacto operacional (23/07/2026, UTC)

- 23:23:31 - enviamos uma mensagem GEN0001 ao BACEN pelo SPB.
- 23:24:52 - nosso supervisor de operações marcou a operação como erro por timeout: a resposta GEN0001R1 chegou pela MQ, mas a decifra no HSM falhou com 500/504.
- 23:30:44 - nova falha de decifra na reentrega (3285 ms); às 23:30:46 a decifra finalmente completou em 98 ms, quando a operação já havia sido encerrada com erro.

O mesmo padrão explica lentidão intermitente de 1 a 6 segundos que medimos em envios de pagamento quando uma assinatura ou um handshake pega a janela ruim.

4. O que já verificamos e ajustamos do nosso lado

Inventário completo das chamadas da nossa aplicação ao HSM: não existe nenhuma rotina que faça sign, decrypt ou get-session em laço. A sessão é cacheada por 5 minutos (N operações consomem 1 get-session). A única chamada periódica é um GET /v1/health a cada 20 segundos, que não toca o vHSM. Em repouso, o SPB não emite nenhuma chamada criptográfica.

Além disso, na noite de 23/07 identificamos que a maior parte do nosso consumo de sign-rsa vinha de handshakes mTLS com o BACEN (conexões novas), e corrigimos: passamos a manter conexões persistentes. Resultado medido: de aproximadamente 24 mil sign-rsa por dia para menos de 2 mil por dia, uma redução acima de 90%. Mesmo com o nosso volume reduzido a menos de um décimo, os 504 continuam ocorrendo na mesma proporção por chamada, o que reforça que a causa não é o nosso consumo.

Também implementamos retry com backoff e jitter para os status transitórios. Isso reduz falha final, mas não resolve a causa: cada 504 custa 3 segundos, e como o nosso sistema opera abaixo de 500 ms por transação não há orçamento de tempo para absorver esse custo.

5. O apoio que precisamos da RTM

1. Validar nos logs do gateway EcosCryptoServer e do vHSM 60042 as ocorrências listadas na seção 3 (os timestamps estão em UTC) e nos orientar sobre a causa observada do lado de vocês.
2. Elevar, ou tornar configurável para a Monetarie, o HttpClient.Timeout de 3 segundos entre o gateway e o vHSM, ou sanar a lentidão do backend que faz as respostas ultrapassarem esse limite.
3. Investigar por que o GET /v1/health chega a levar cerca de 3 segundos sem executar operação criptográfica.
4. Nos informar os limites de concorrência e de taxa suportados pelo vHSM 60042, para dimensionarmos nosso pool de conexões sem induzir timeout. Estamos em rampa de aumento de volume e queremos planejar essa capacidade junto com vocês.

6. Disponibilidade

Podemos reproduzir a medição da seção 3.1 sob demanda, em janela combinada, e acompanhar em conjunto qualquer teste que o time da RTM queira conduzir nos dois ambientes. Ficamos à disposição para uma call técnica.

Equipe técnica Monetarie SCD