

# REENGENHARIA DE APLICAÇÕES ASP.NET LEGADAS PARA ARQUITETURA DE SERVIÇOS

Metodologias, processos e  
técnicas

# YAN JUSTINO



I'm a Software Engineer living in Natal/Brasil. Also, I am a master degree in Software Engineering at UFRN/IMD.

yanlimaj@gmail.com

MSC, SOFTWARE ENGINEER



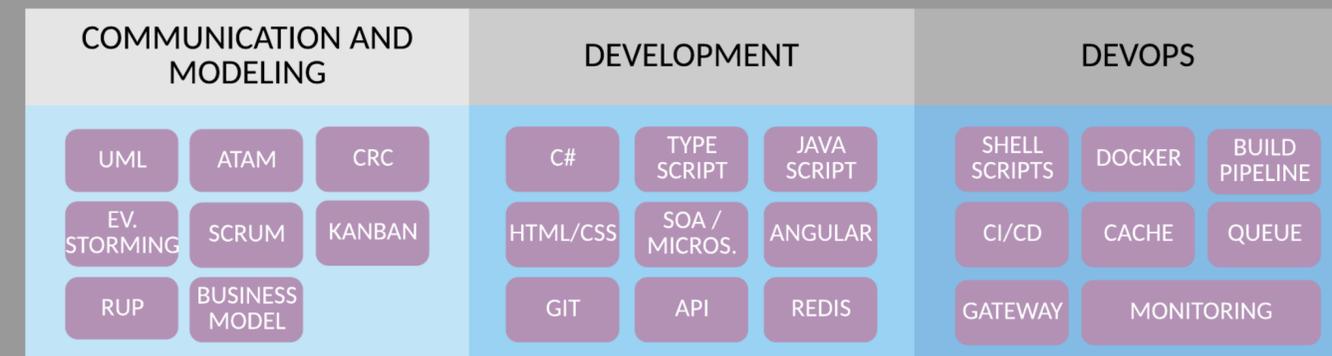
## MY CAREER



## METHODOLOGIES, PROCESS AND PRACTICES



## SKILLS



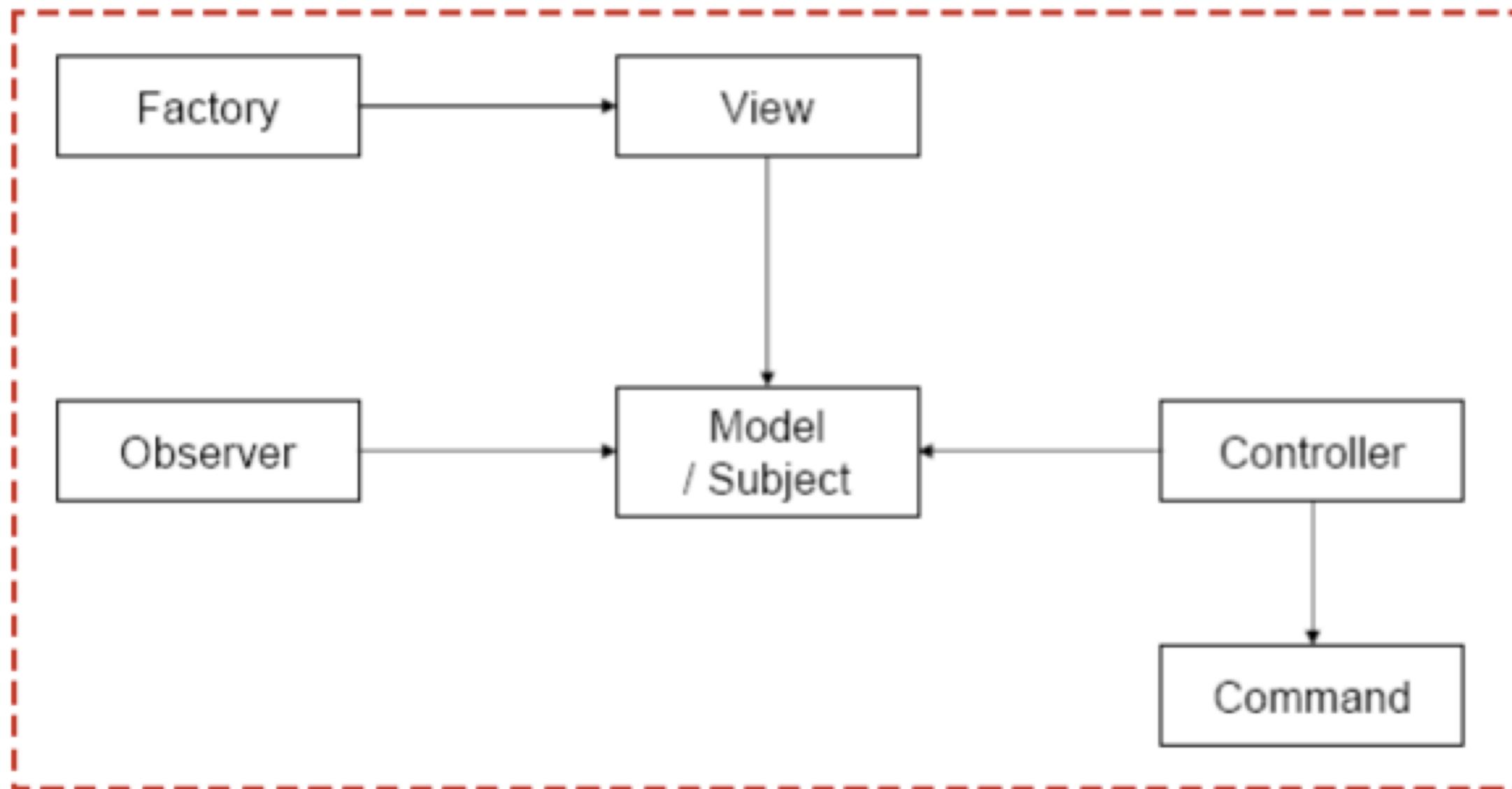
## CERTIFICATIONS AND AWARDS

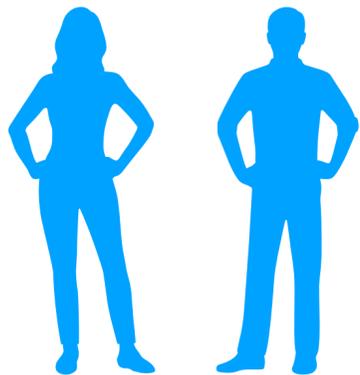
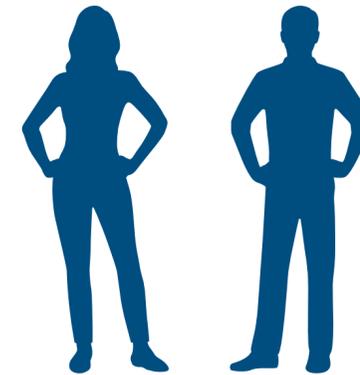
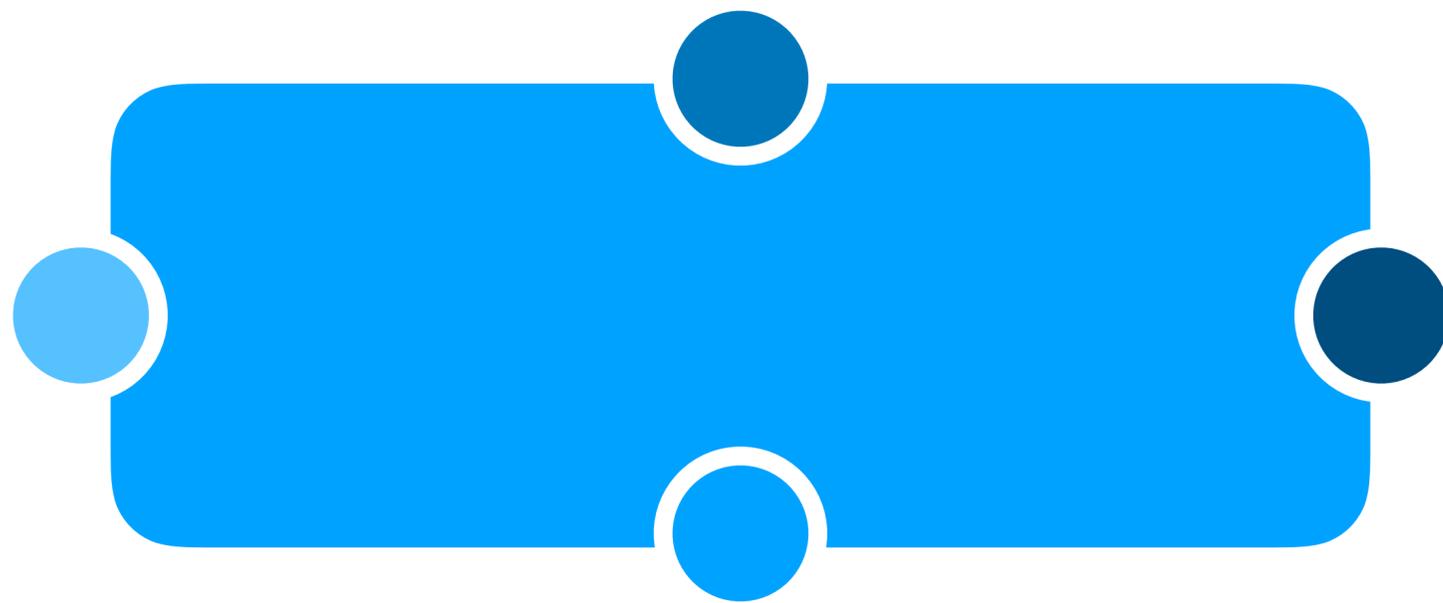
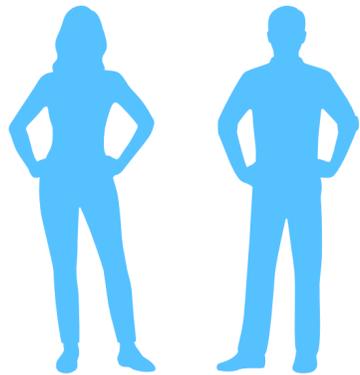
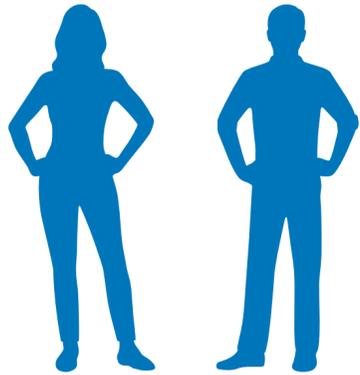


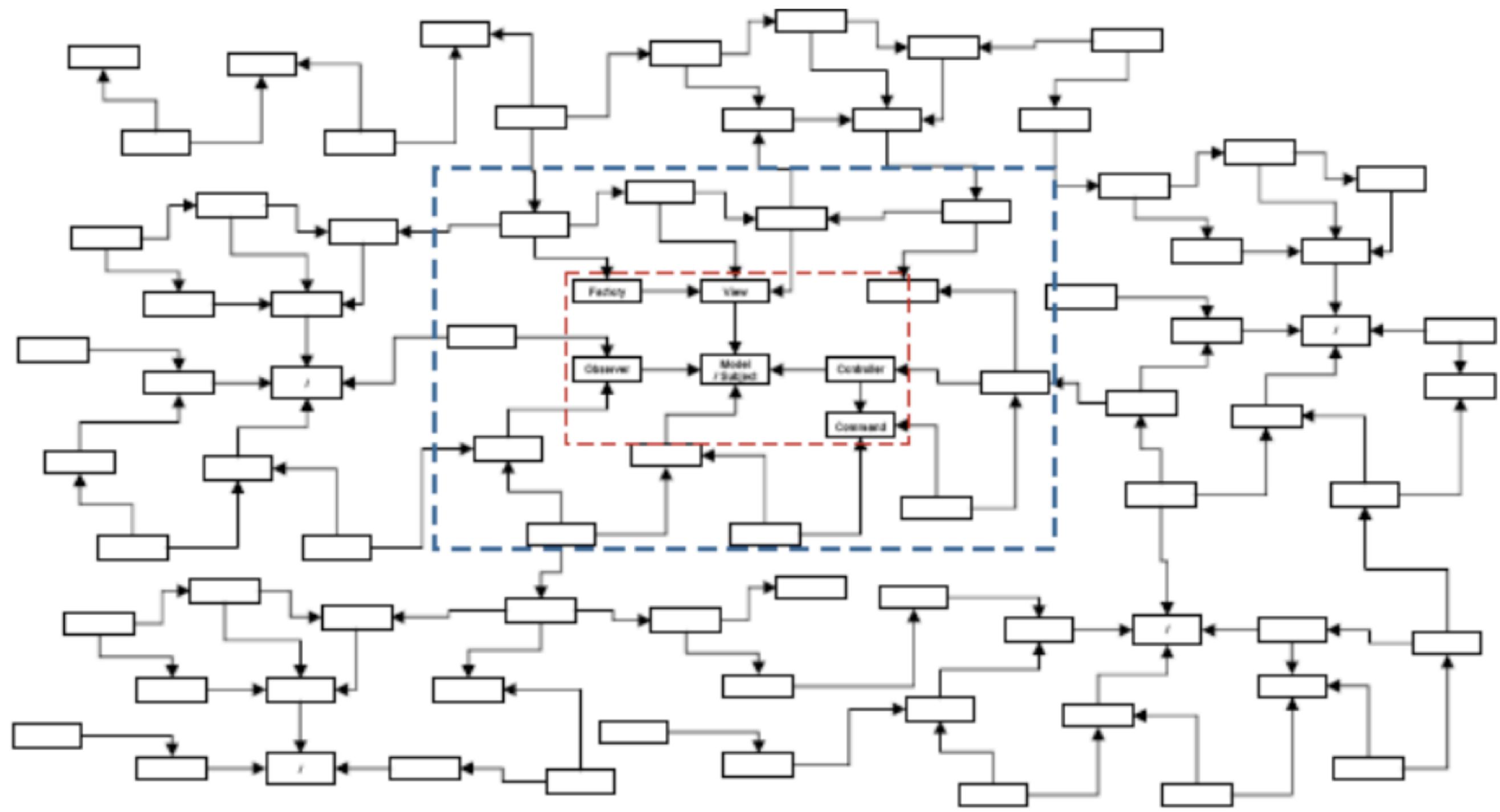
# **EVOLUÇÃO DE SOFTWARE**

**“É a capacidade de alterar, de forma rápida e confiável, um sistema de software para adaptá-lo às mudanças do ambiente.”**

*-(BENNETT, 2000 )*







# **SISTEMAS LEGADOS**

**“Quando um software não é mais viável, é considerado envelhecido,  
em decaimento ou legado.”**

*-(BENNETT ; RAJLICH, 2000 )*



1

### APRESENTAÇÃO

Processamento de UI no servidor

2

### NEGÓCIO

Modelos de negócio compartilhados por vários contextos (acesso irrestrito)

3

### DADOS

Modelos baseados em Entidades Relacionais e Anêmicos

4

### ESTADO DE SESSÃO

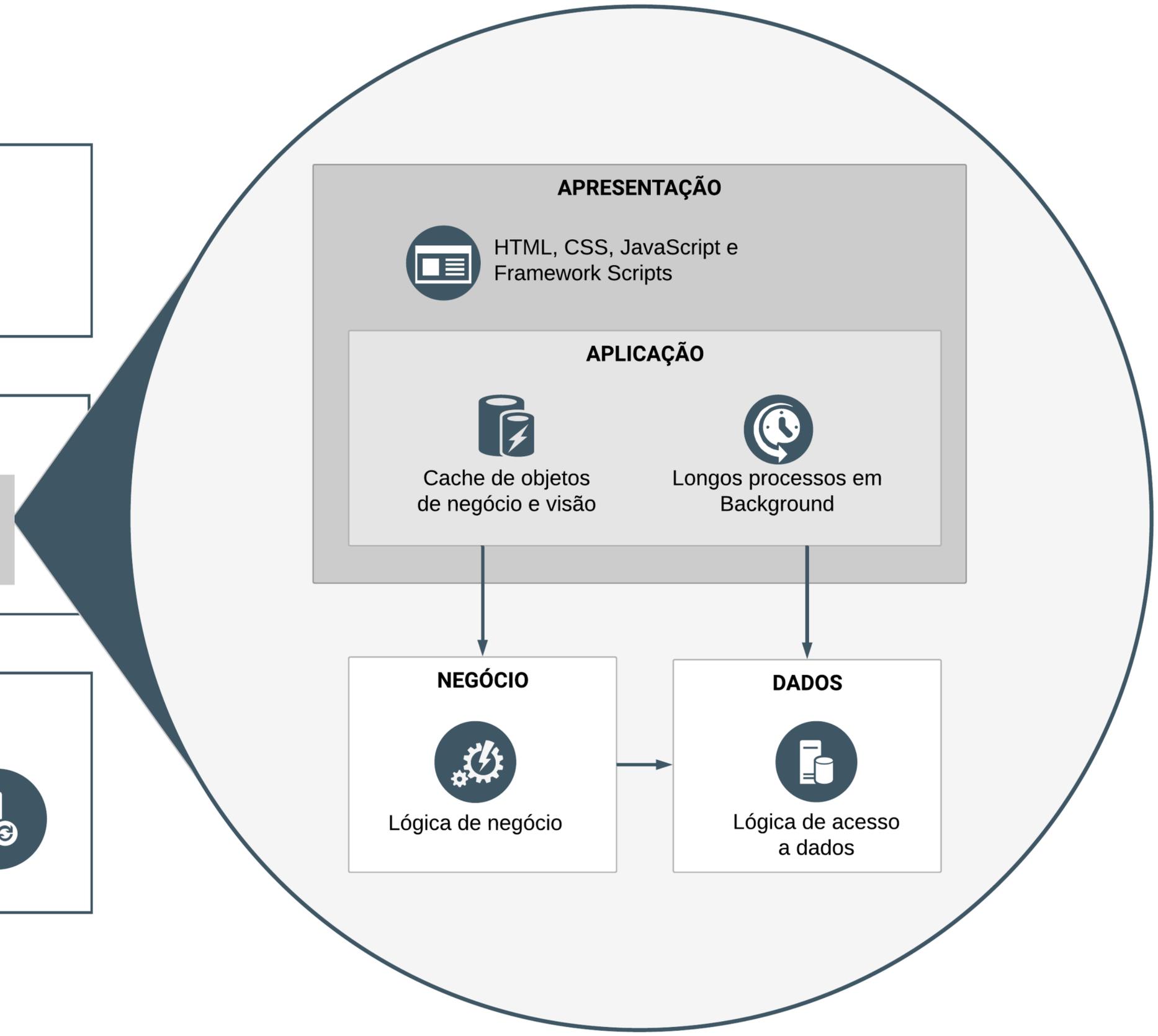
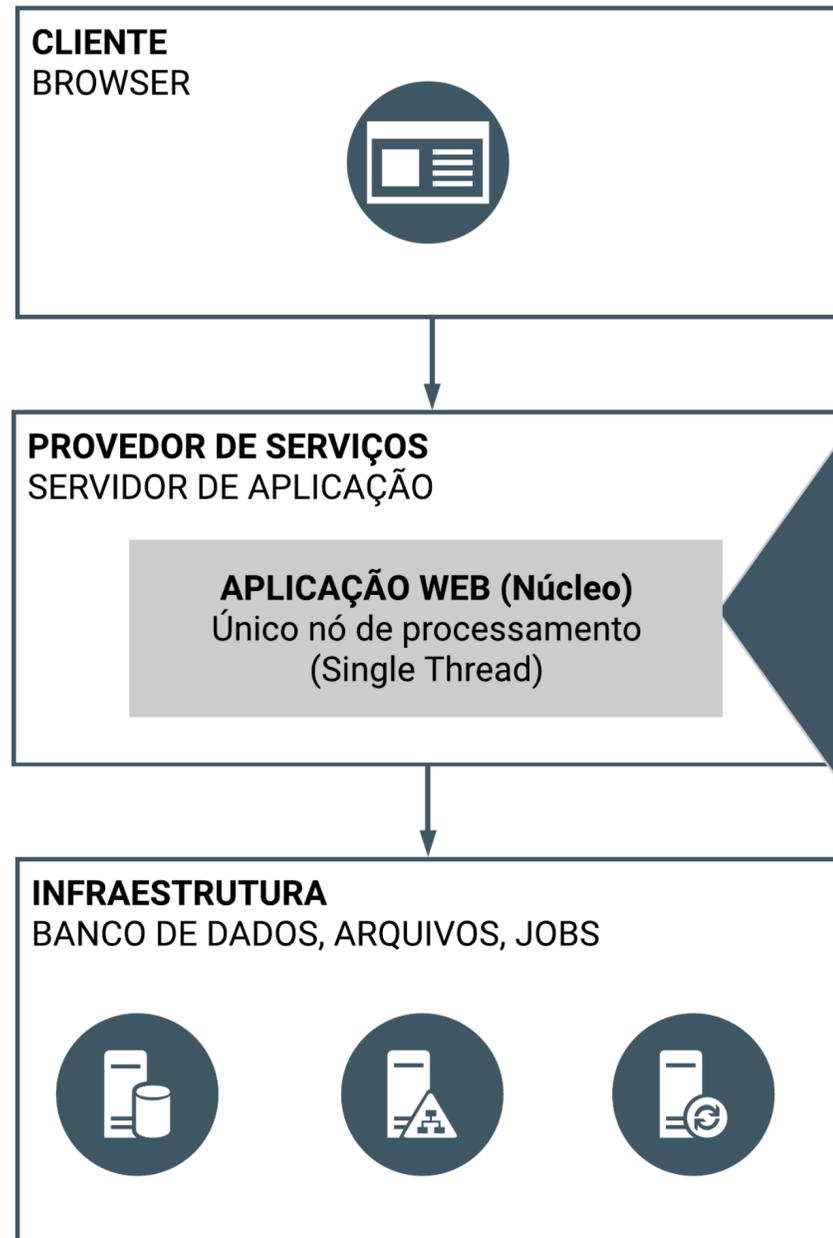
Manutenção de estado de objetos complexos na sessão da aplicação

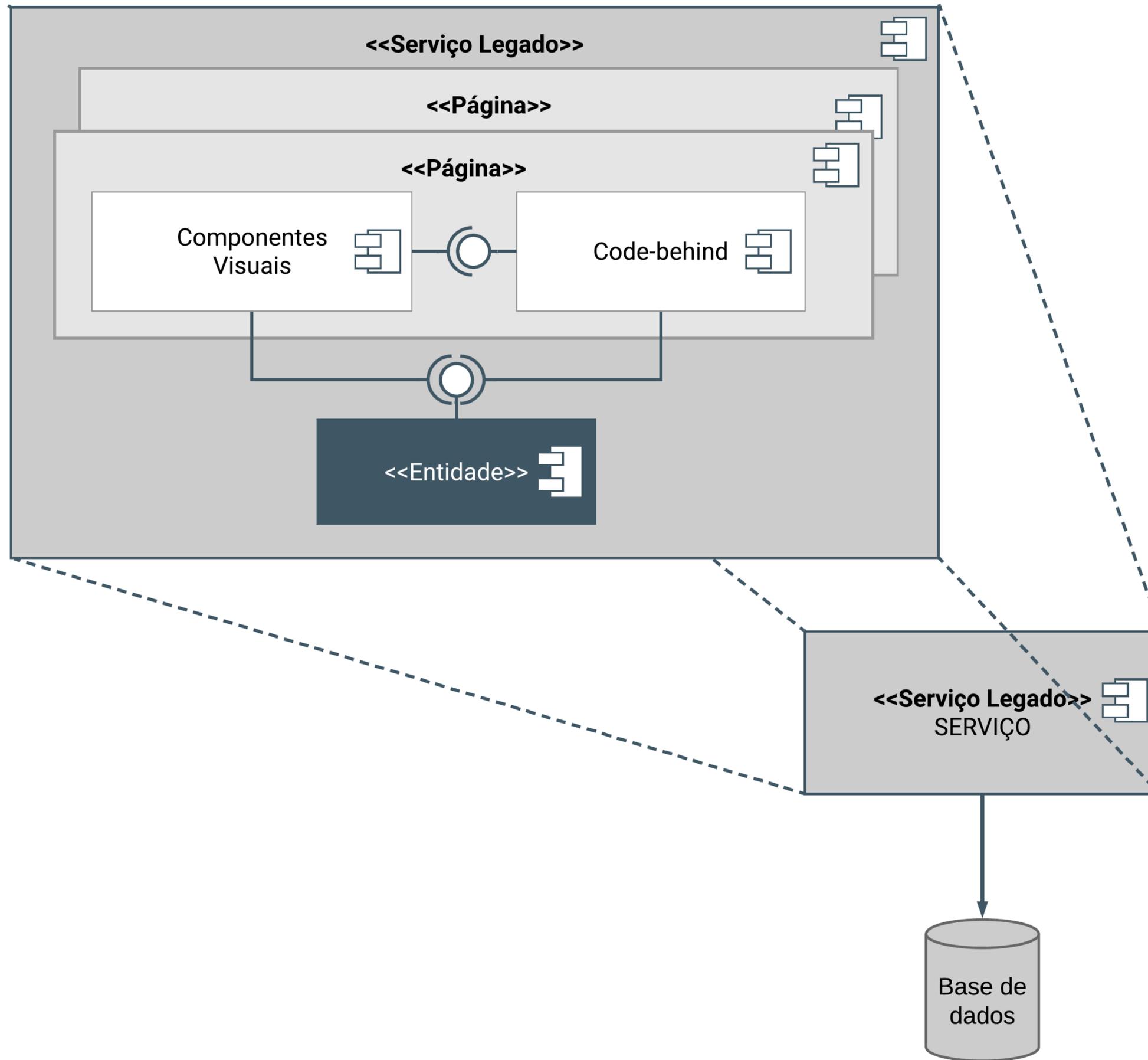
5

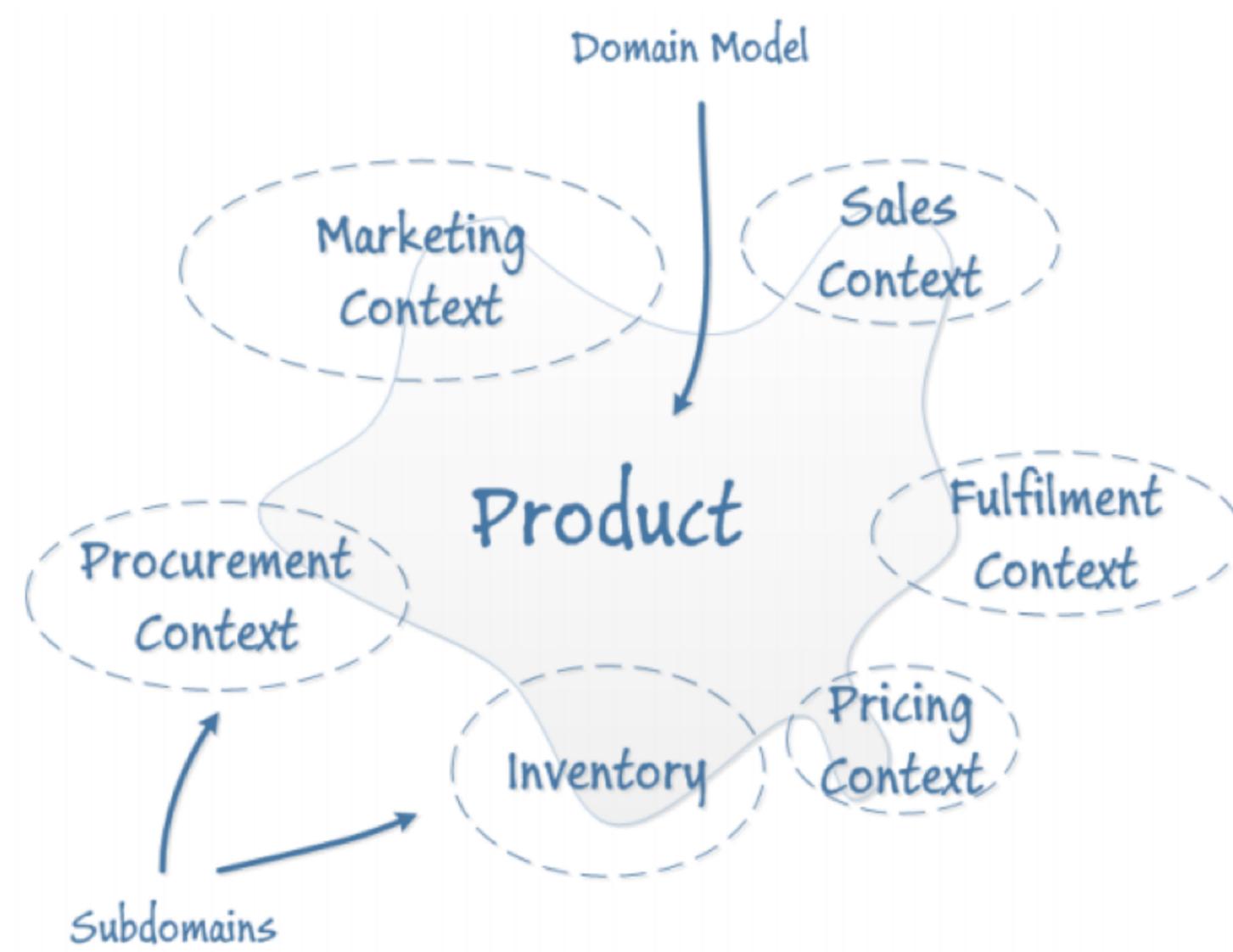
### PROCESSAMENTOS

Alto processamento de dados em cima do processamento principal da aplicação

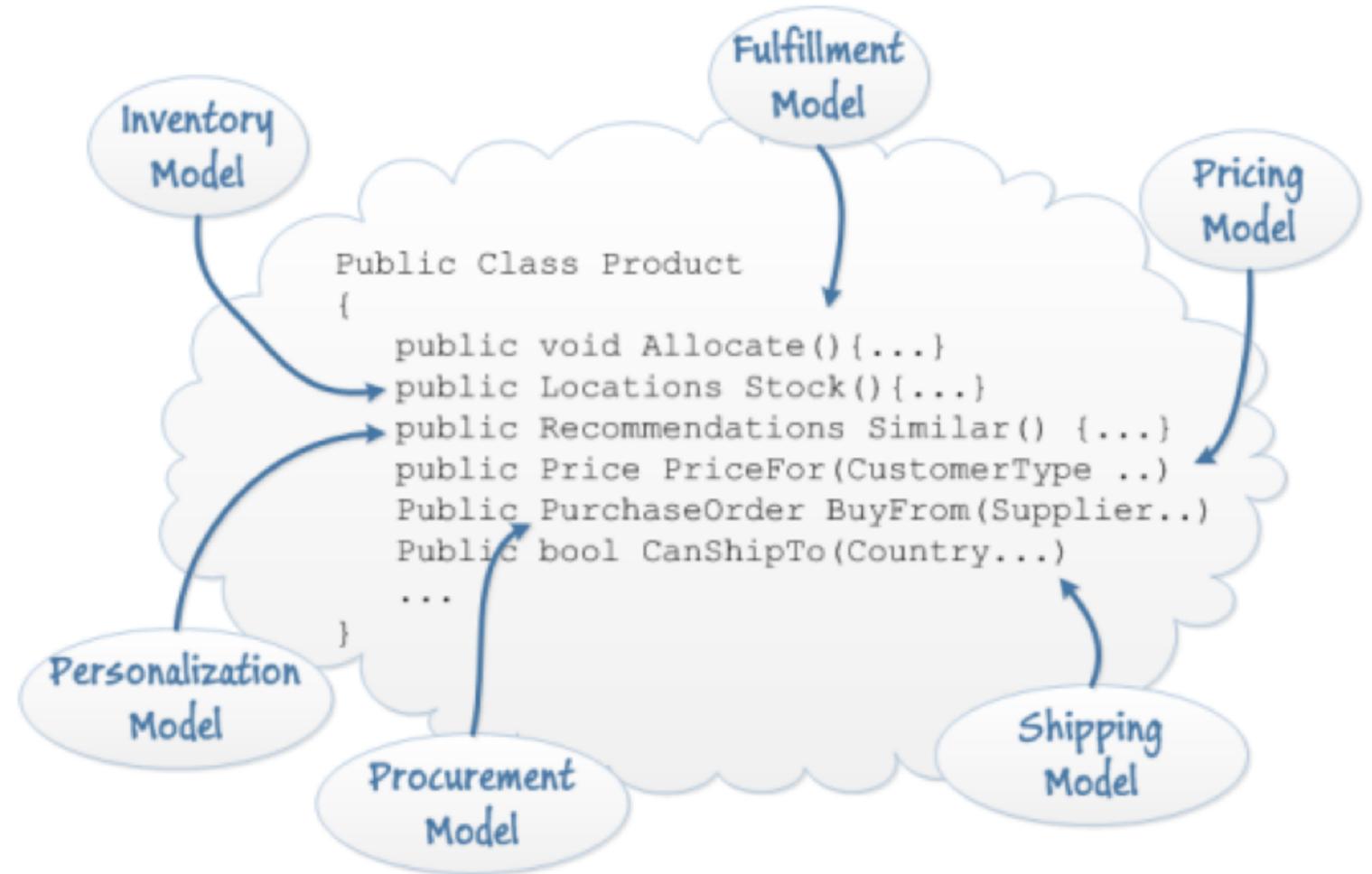
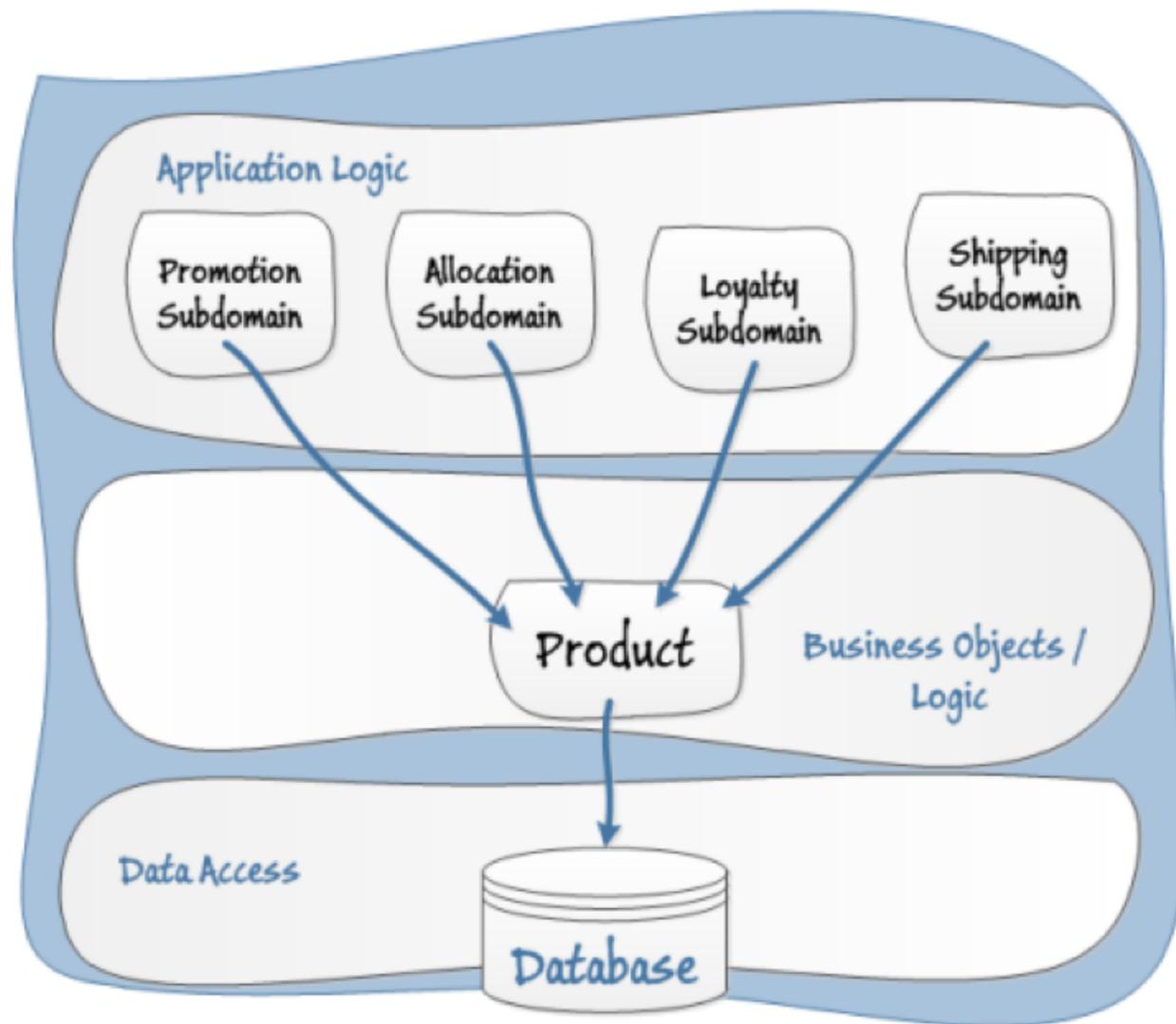
MONOLITHIC  
LEGACY  
SYSTEM

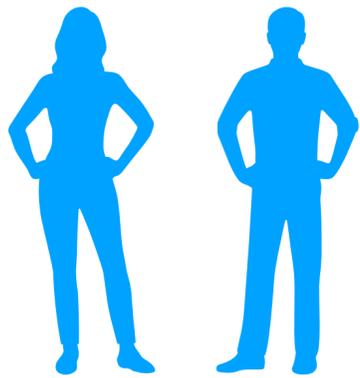
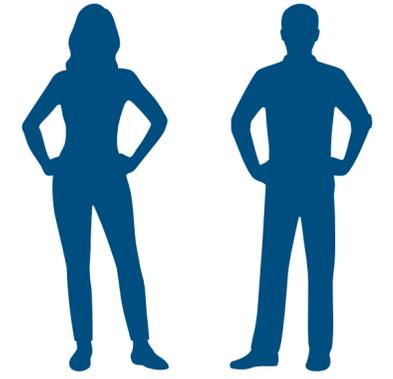
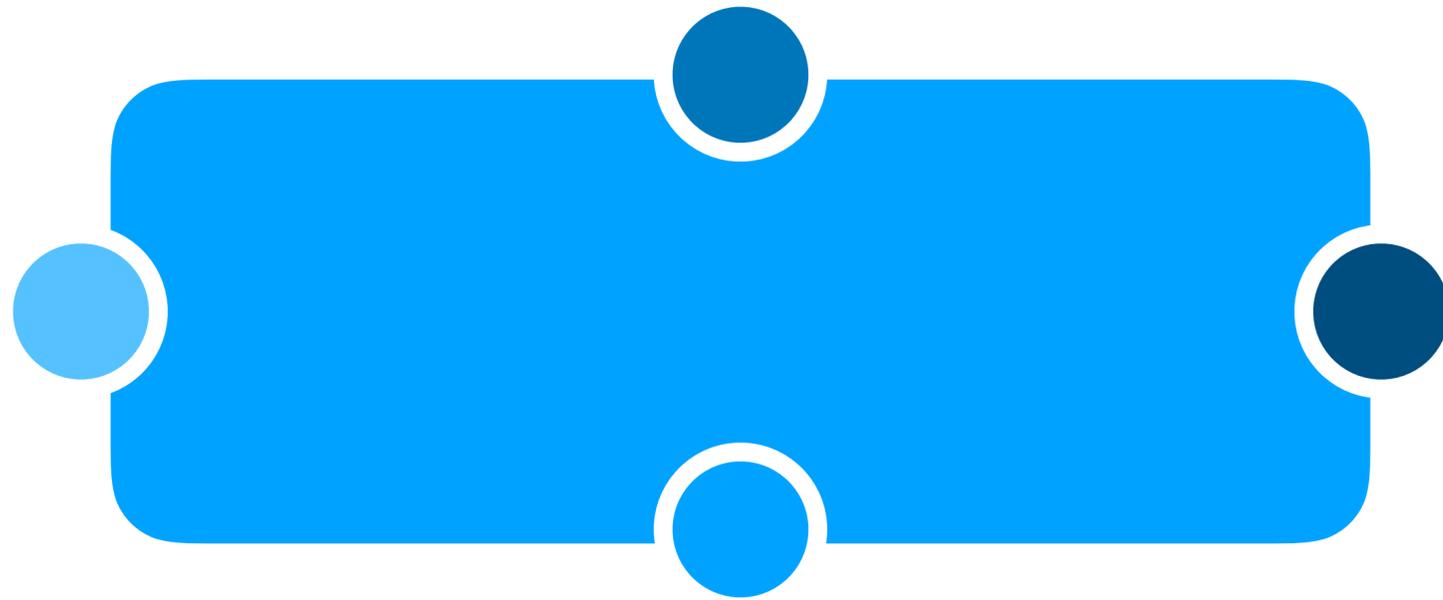
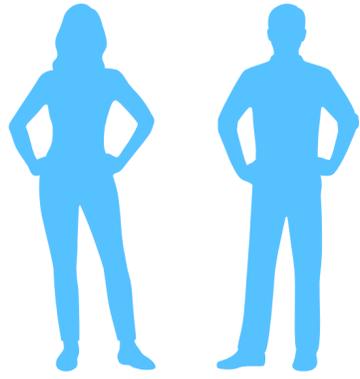
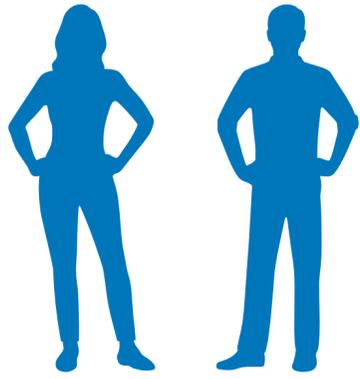




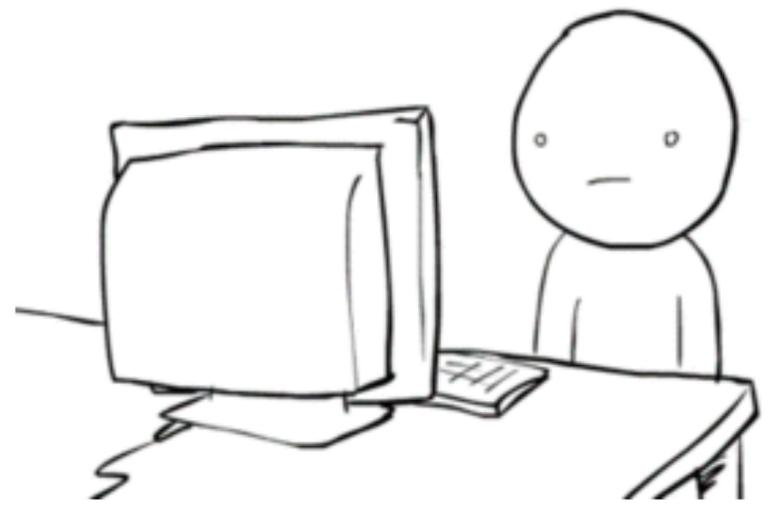


## E-commerce Application





```
8  [-] <Serializable()> _
9  Public Class Debito
10
11  [+ Campos
55
56  [+ Propiedades
494
495  [+ Constructores
2454
2455  [+ Métodos
2965
2966 End Class
```



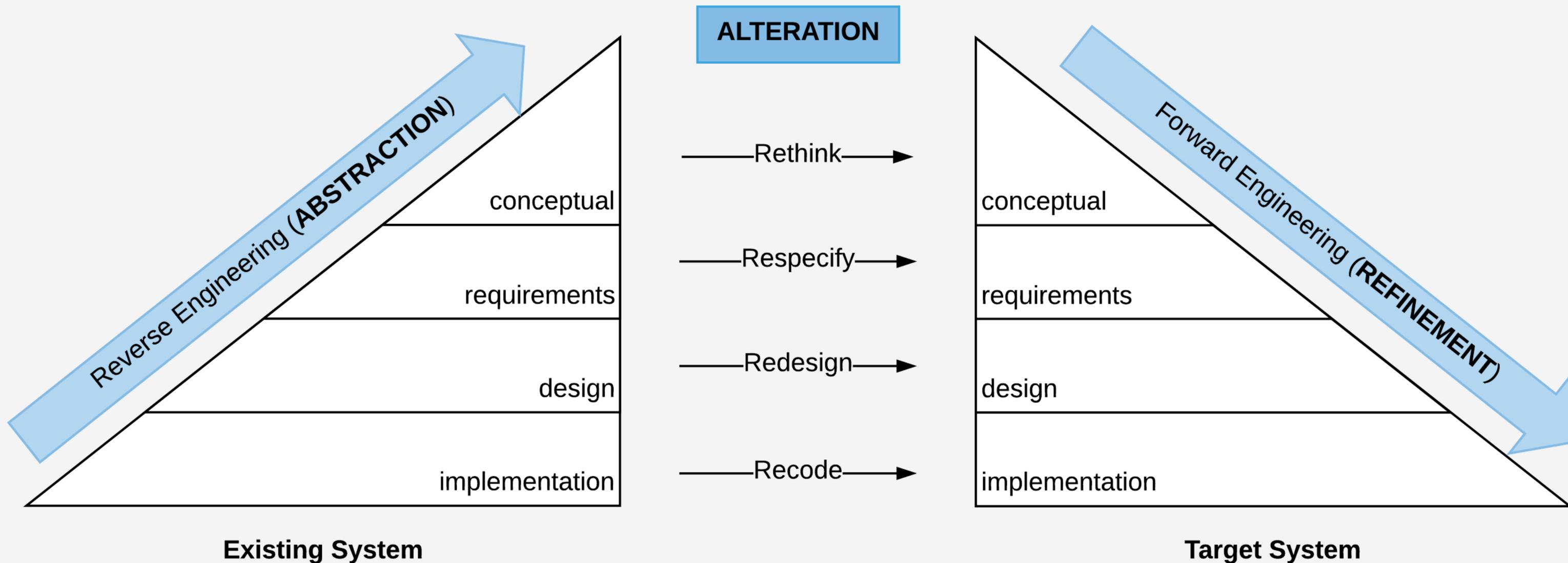
```
...
495  #Region "Construtores"
496
497  + Vazio
504
505  + ICMS Antecipado - Long, Int64, String, Date, Boolean
757
758  + Nota Fiscal Avulsa - Long, Int64, String, Decimal, DateTime, Date
862
863  + Pendencia de Posto - Long, Int64, String, Decimal, Date, Char
990
991  + GIM/FECOP - Long, Int64, String, String, String, Decimal, Integer, Date
1090
1091  + Parcelamento - Long, Int64, Date, Integer, Date
1175
1176  + Simples Avulso - Long, String, String, String, Date, Date, Decimal
1261
1262  + Cresce - Long, Int64, DateTime, Decimal, Date, Boolean
1341
1342  + FECOP Avulso - Long, String, String, String, Date, Date, Decimal
1414
1415  + Alcool - Long, Int64, String, Date, Integer
1506
1507  + ICMS Frete - Long, String, Date, Decimal, Decimal, Decimal, Decimal, Decimal, String, Decimal, String, String, String
1629
1630  + TADF Alcool - Long, String, String, String, Date, Date, Decimal, String, Decimal, Decimal, Decimal, Decimal, Decimal, Decimal, String, String
1809
1810  + Alcool Avulso - Long, String, String, Date, Date, Decimal, String, Decimal, Decimal
1887
1888  + Débitos de Saída
2452
2453  #End Region
```

**Muitas vezes, esses sistemas legados são vitais para os objetivos estratégicos das organizações e não podem ser simplesmente encerrados e descartados.**

# REENGENHARIA DE SOFTWARE

**“Oferece uma estratégia de modernização de sistemas pelo atendimento de importantes propósitos como portar sistemas para uma nova plataforma, introduzir novas tecnologias, extrair conhecimentos, design e quebrar arquiteturas monolíticas.”**

*–(WAGNER, 2014)*



PROCESSO DE REENGENHARIA DE SOFTWARE

**“Um dos principais desafios da Reengenharia de Software é garantir a equivalência funcional na nova versão”**

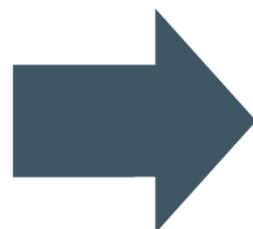
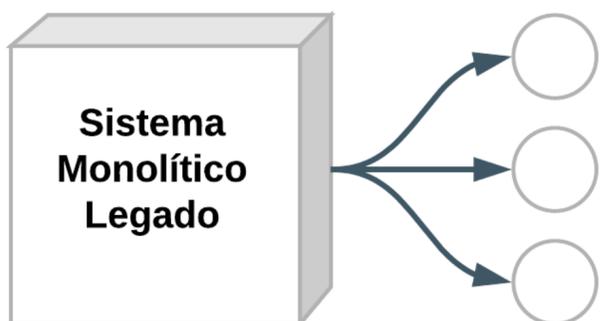
*– (GRUBB; TAKANG, 2003)*

**“Ao passo que um sistema legado entra num processo de Reengenharia de Software, é importante que se trace estratégias para que ele seja migrado para atingir as metas de qualidade desejadas.”**

# 1

## ADOTAR A ORIENTAÇÃO A SERVIÇO

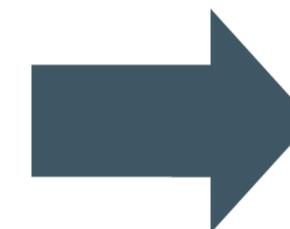
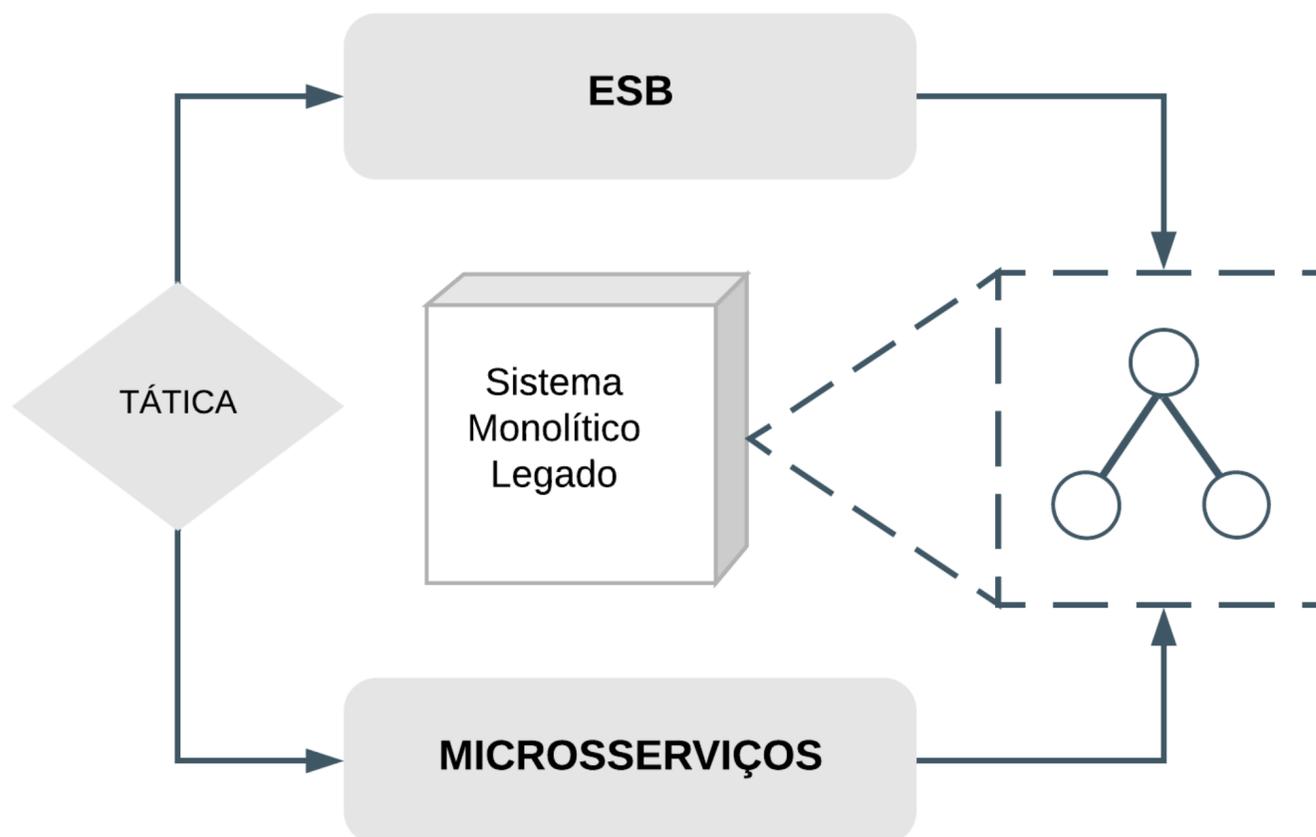
COMO PARADIGMA PARA IDENTIFICAÇÃO E MODELAGEM DE SERVIÇOS



# 2

## APLICAR REENGENHARIA DE SOFTWARE

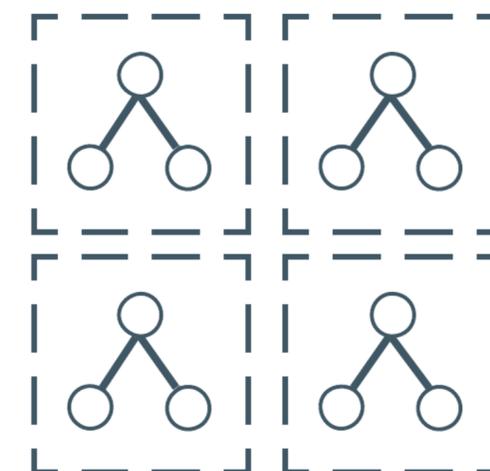
PARA DECOMPOR O SISTEMA MONOLÍTICO E CONSTRUIR SOLUÇÕES ORIENTADAS A SERVIÇOS



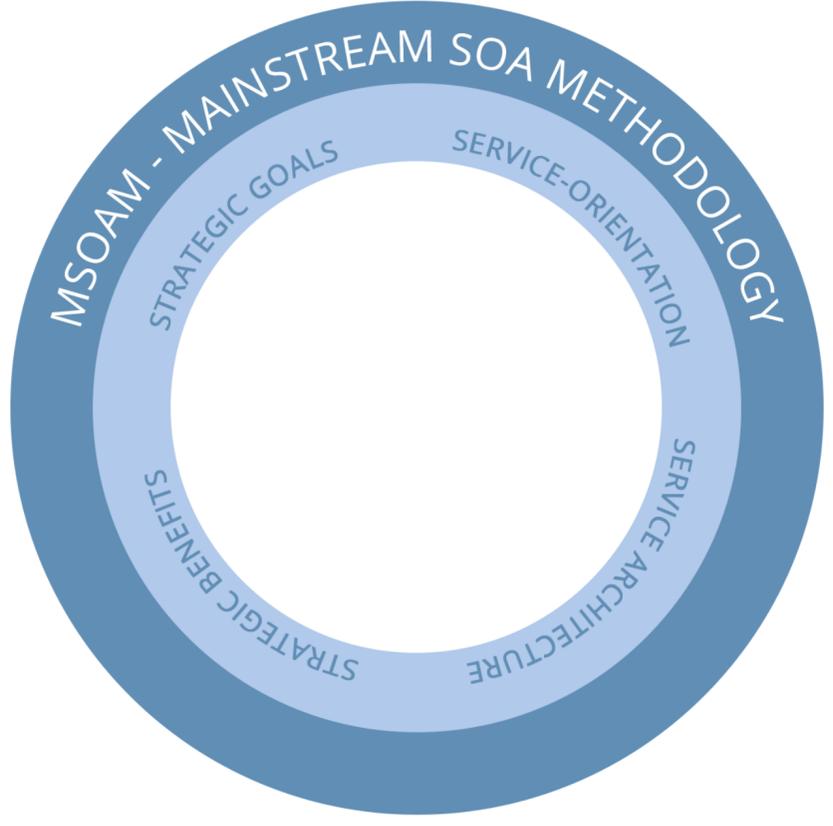
# 3

## ADOTAR PRÁTICAS CONTÍNUAS

PARA MELHORAR A QUALIDADE DE IMPLANTAÇÃO E MONITORAMENTO

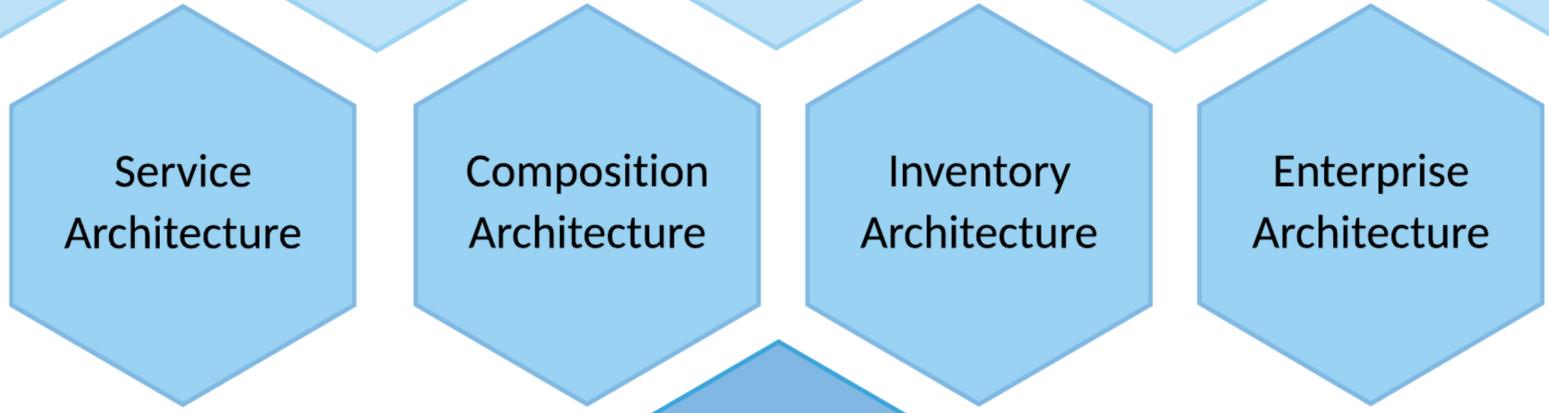
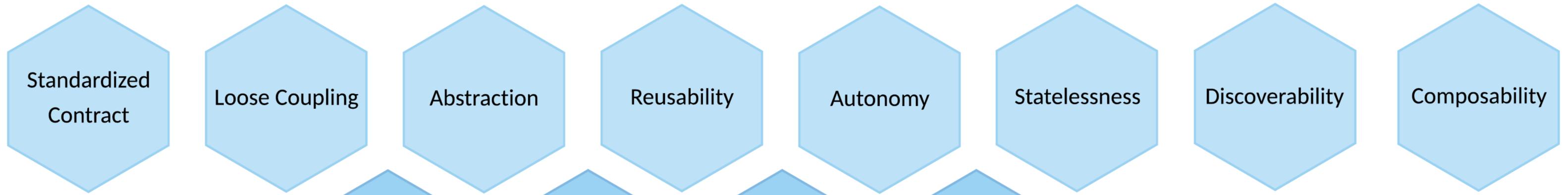


# **ADOTAR ORIENTAÇÃO A SERVIÇO**



**PRINCIPLES**

Service-orientation



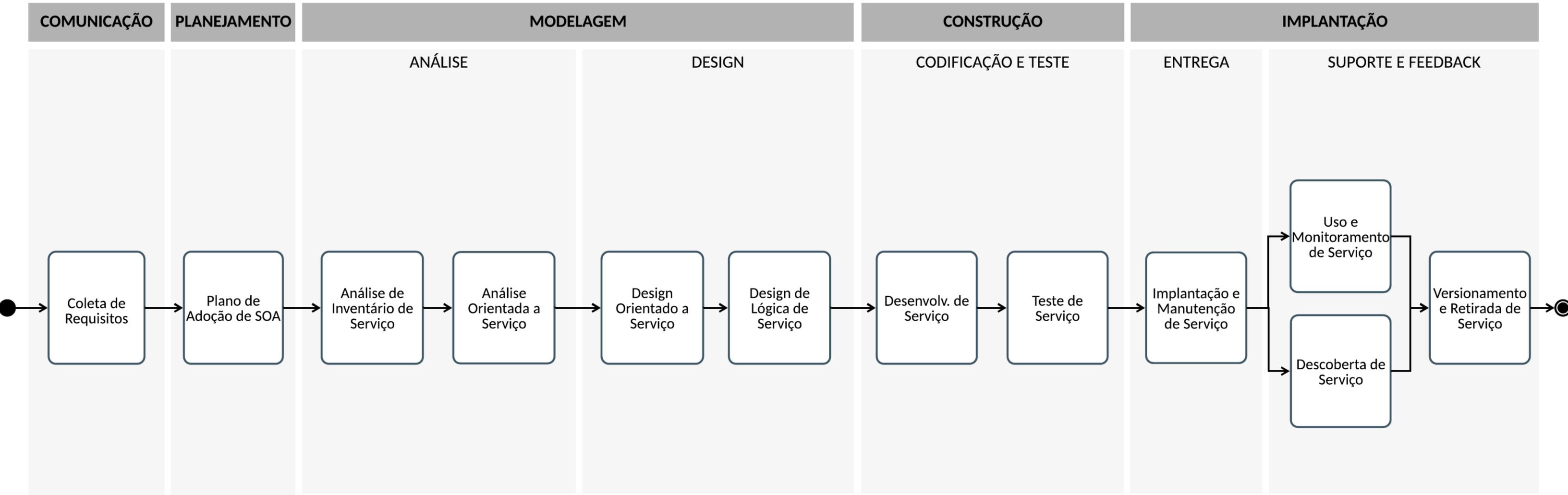
**STRATEGIC**

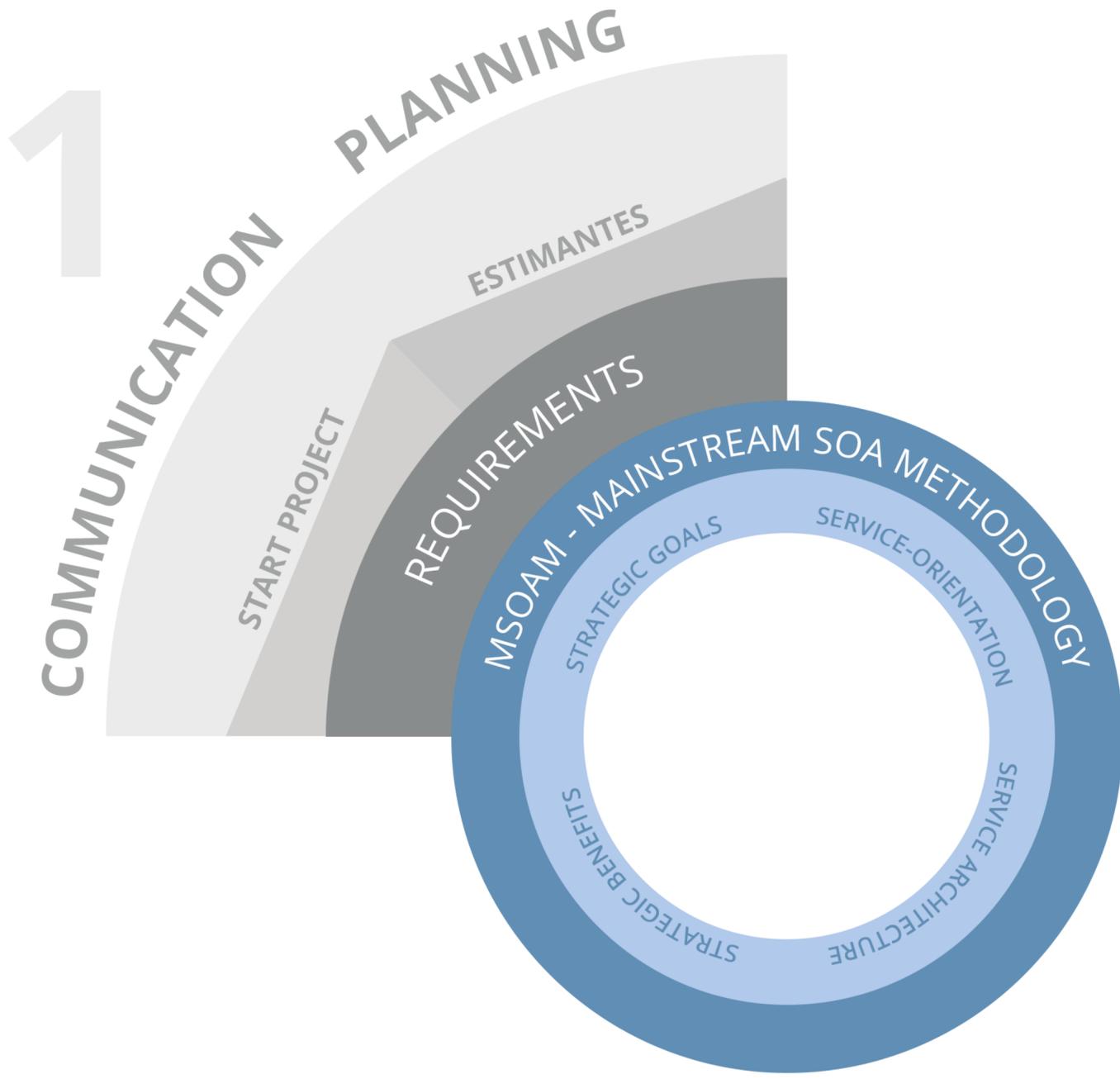
Service-Oriented Architecture

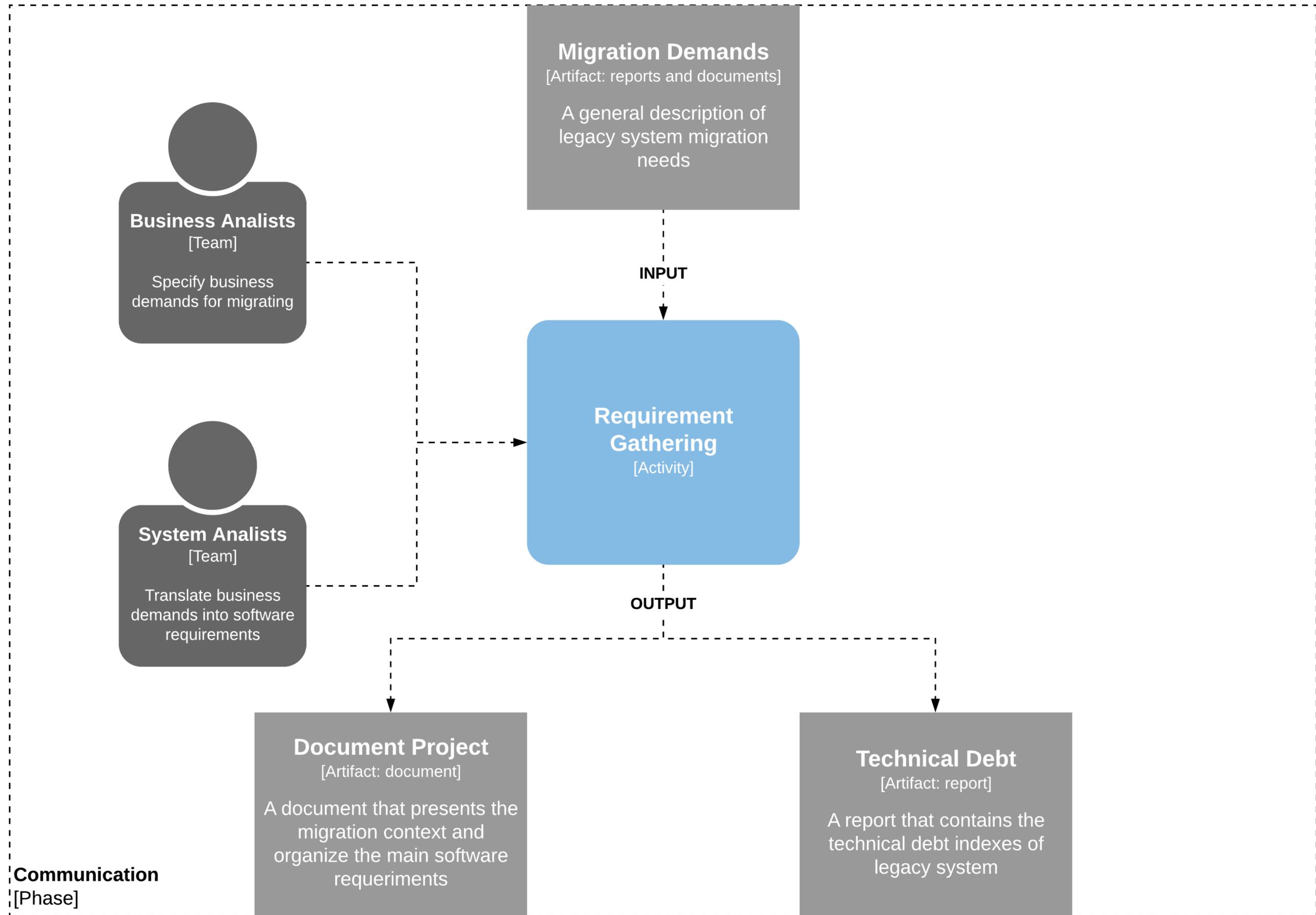


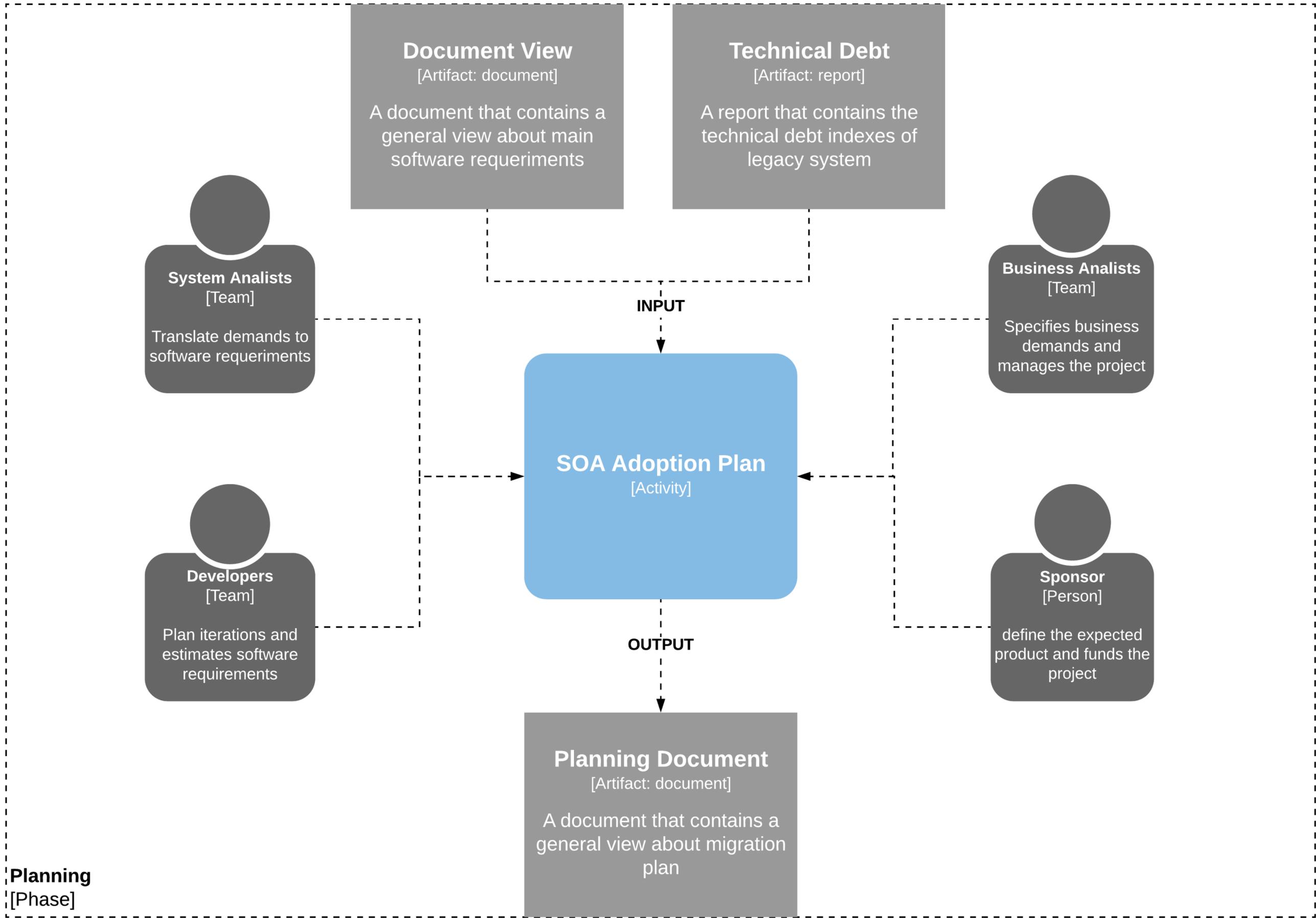
**TACTICAL**

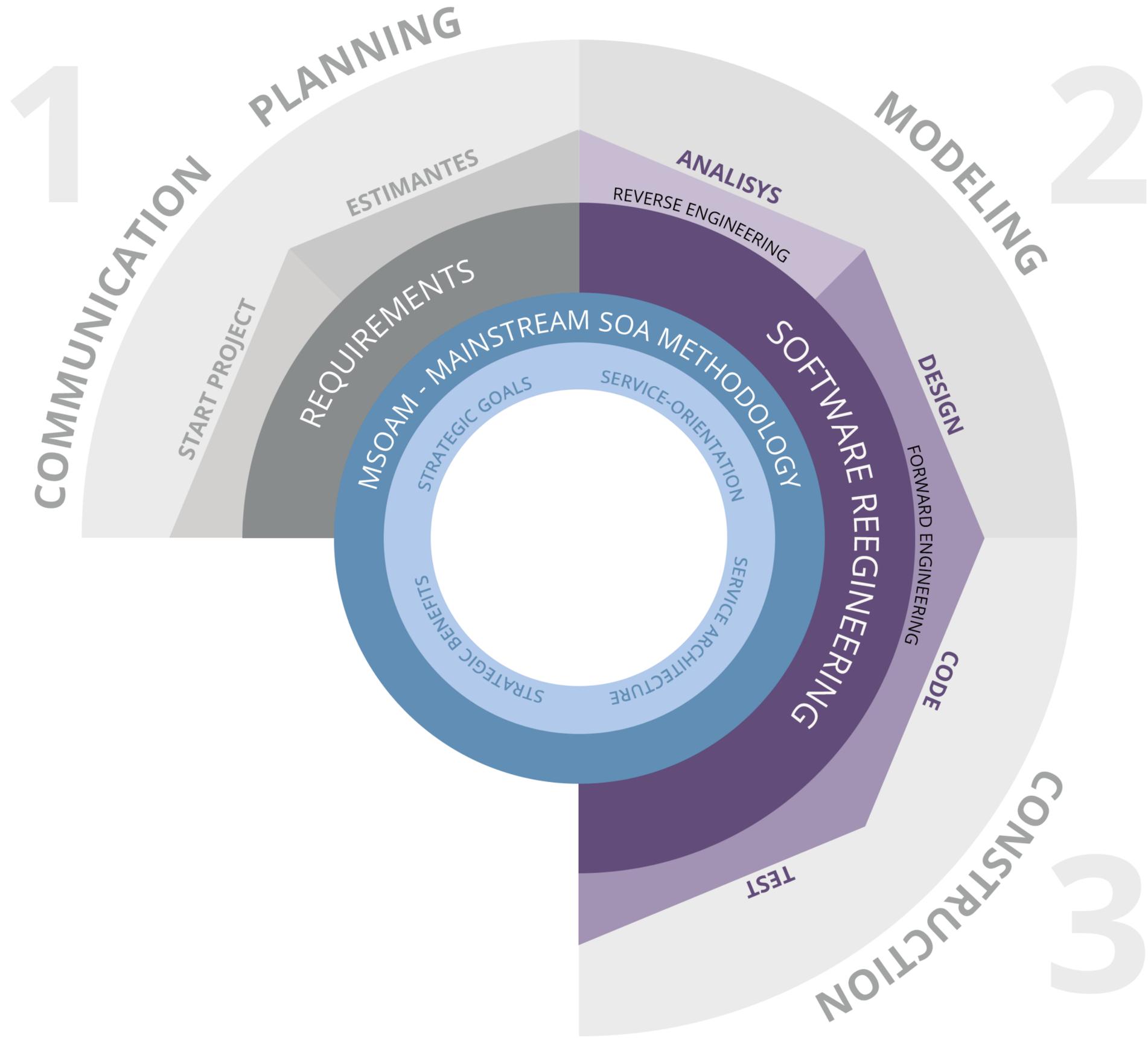
Domain-Driven Design, CQRS & Microservice



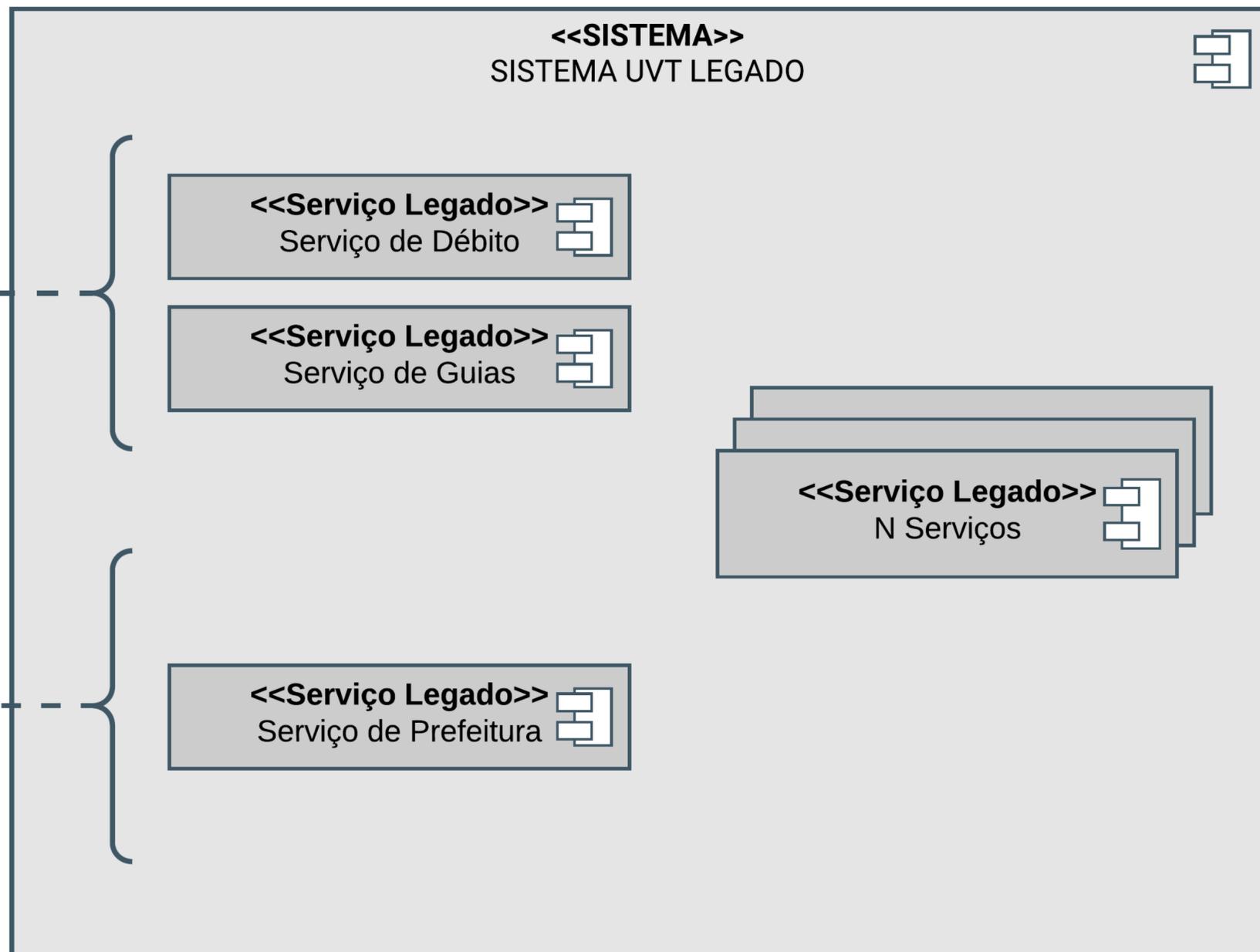
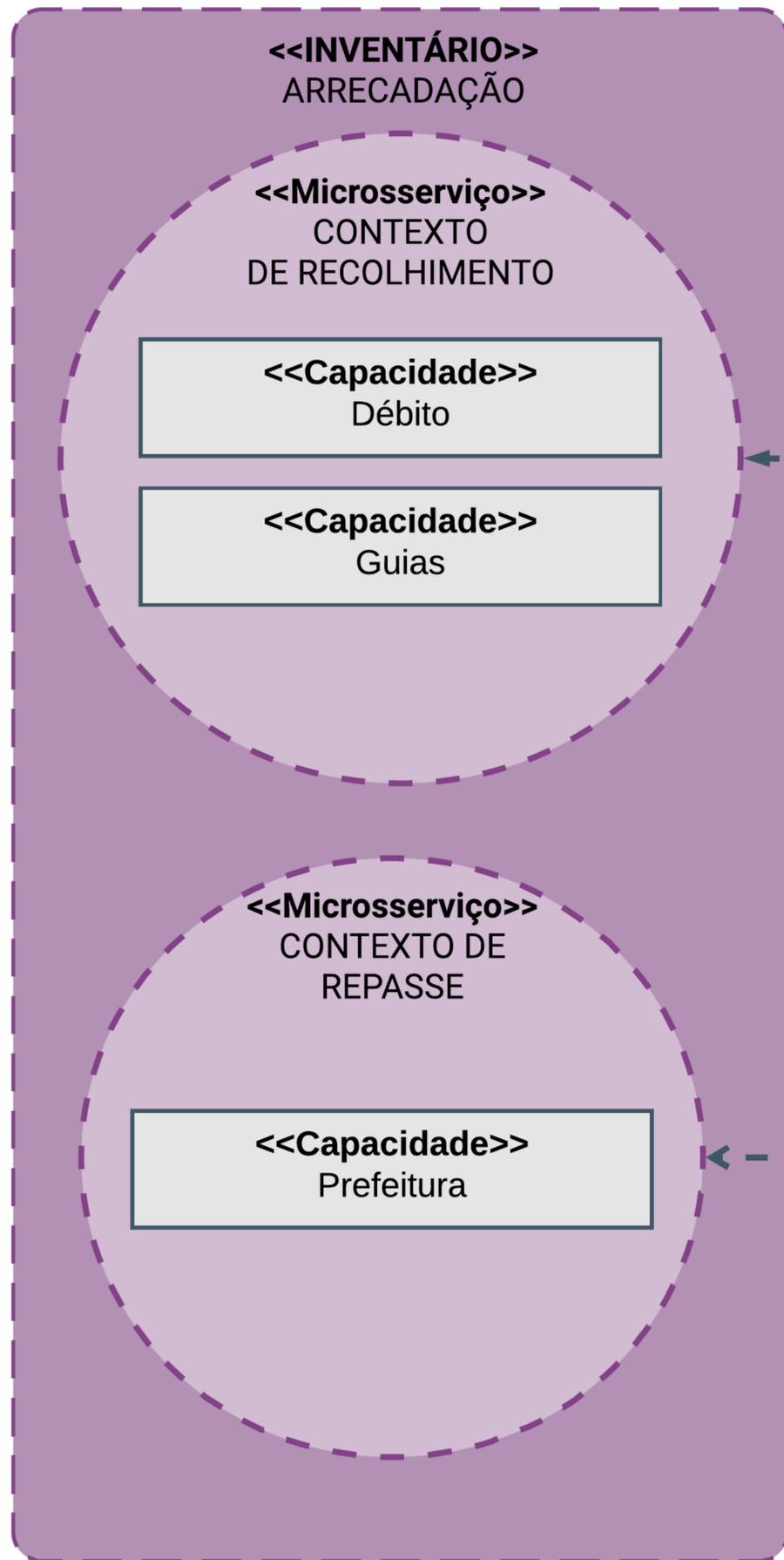


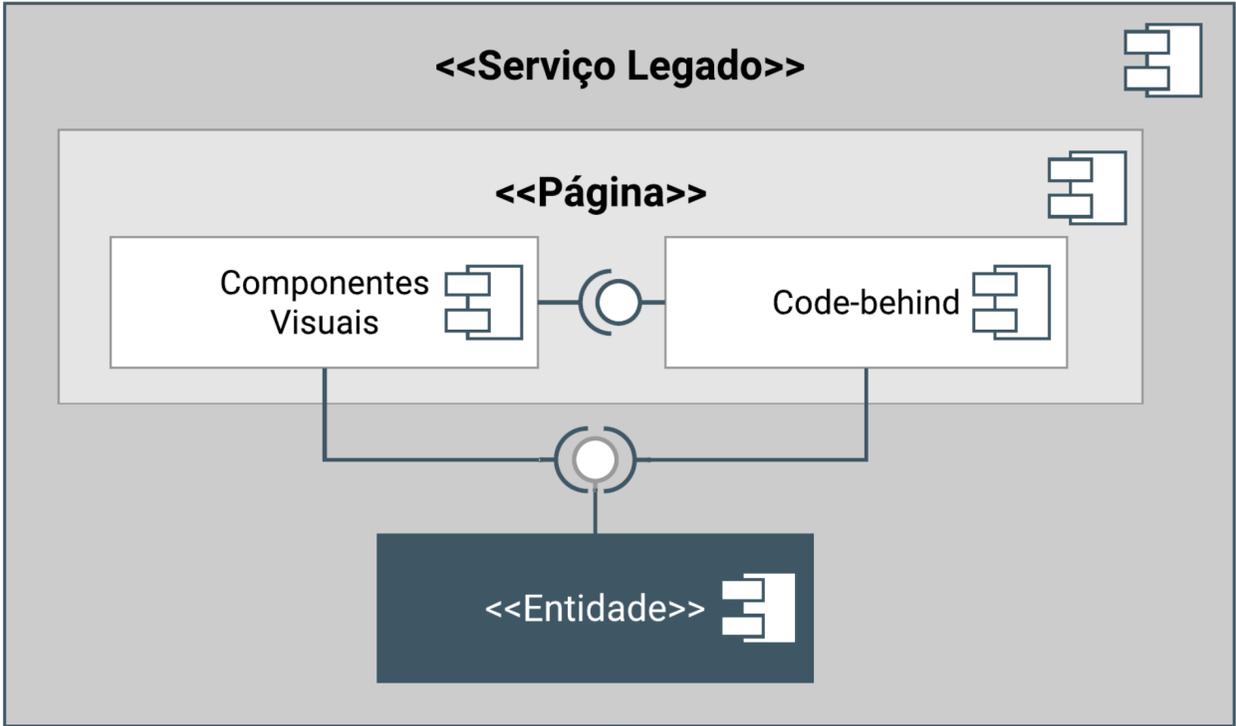




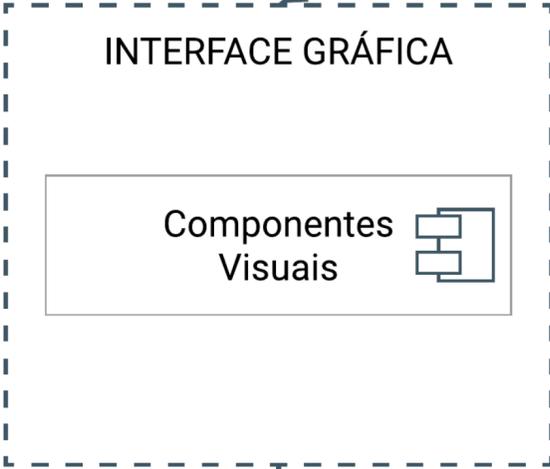


**ANÁLISE**





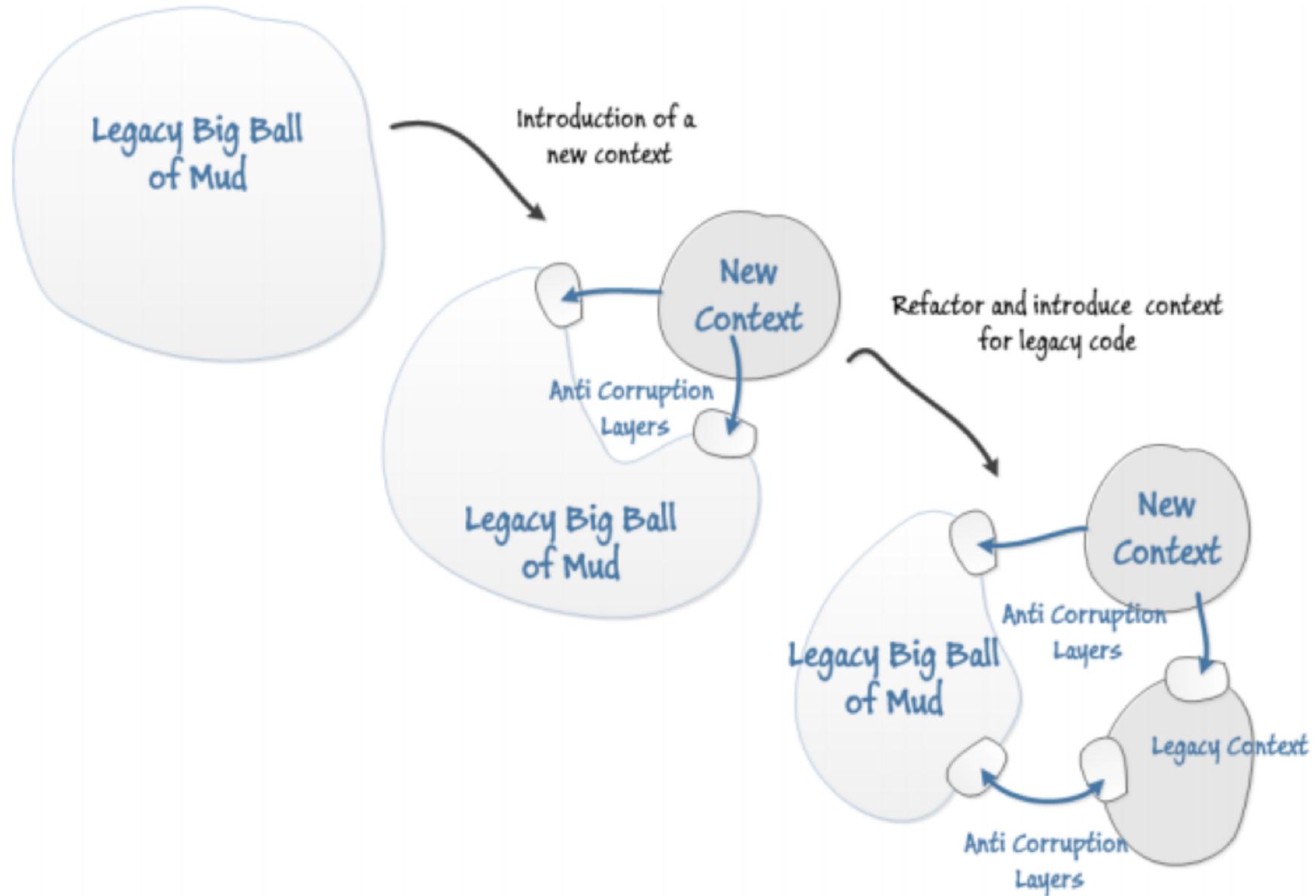
ENGENHARIA REVERSA

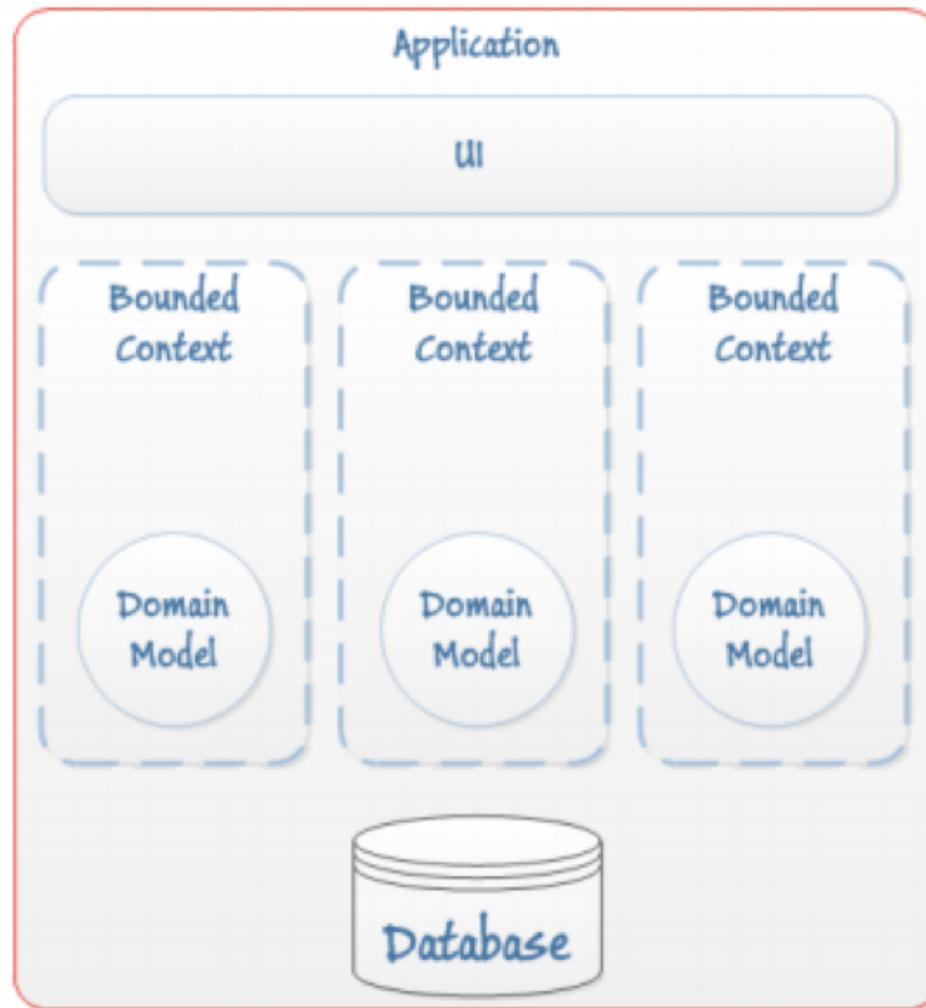


**<<artefatos>>**  
Protótipos, descrição do comportamento da GUI, modelos de visão

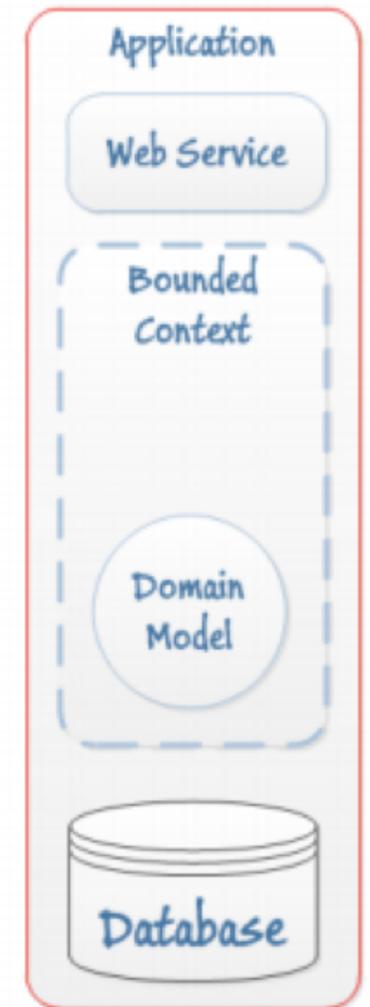


**<<artefatos>>**  
Diagramas de classe, casos de uso, classe (código-fonte)

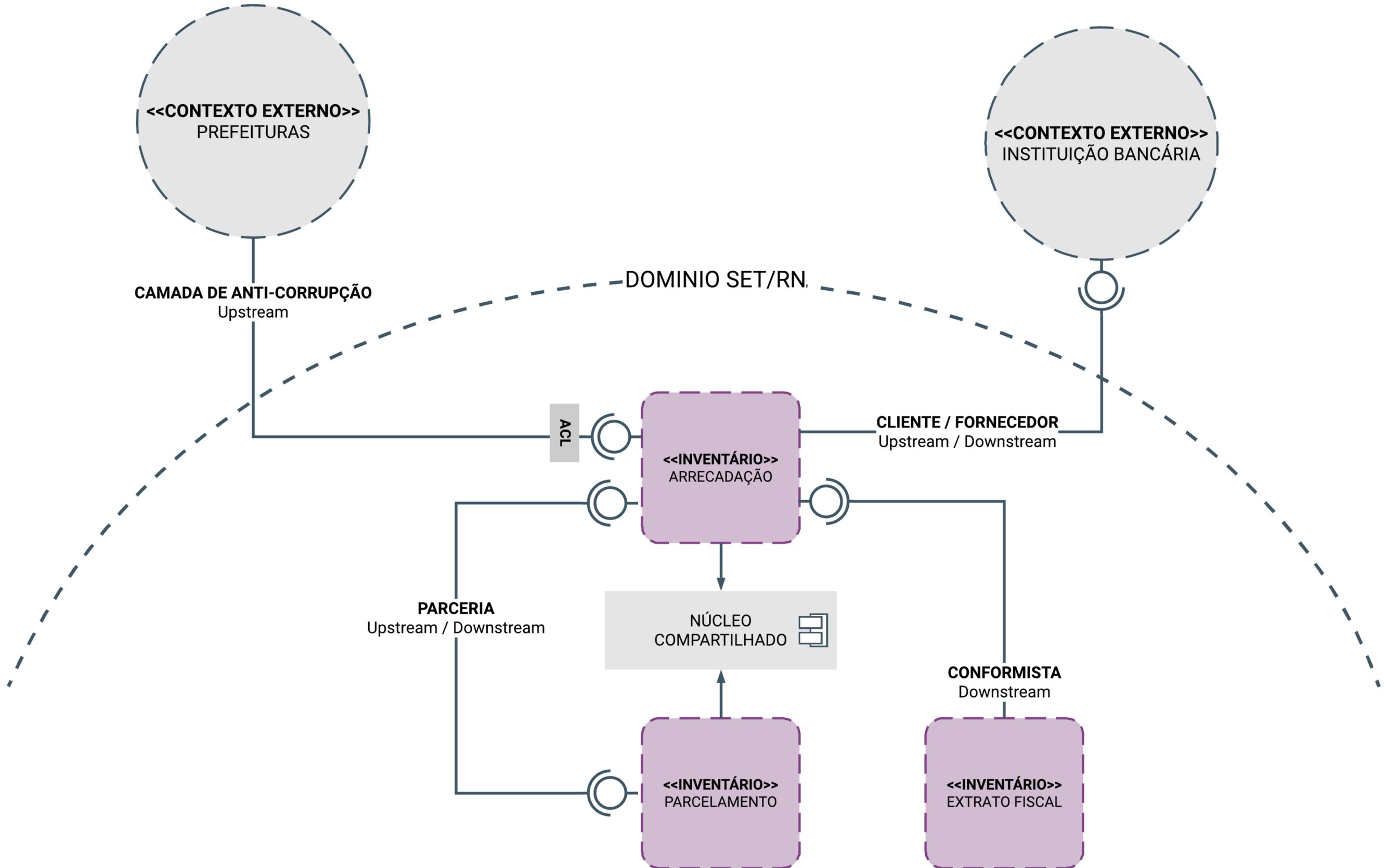




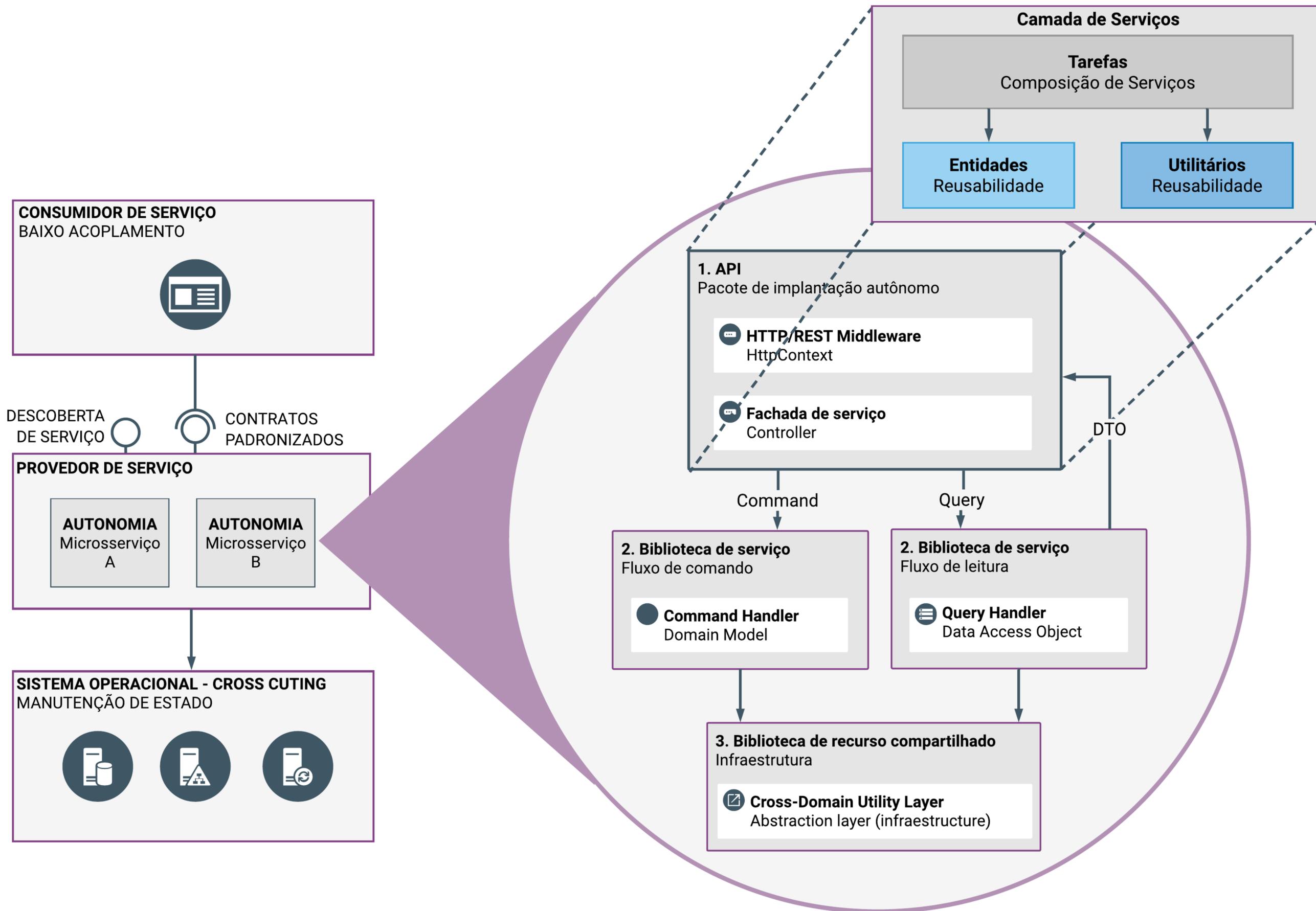
An application made up of three bounded contexts all persisting to the same database

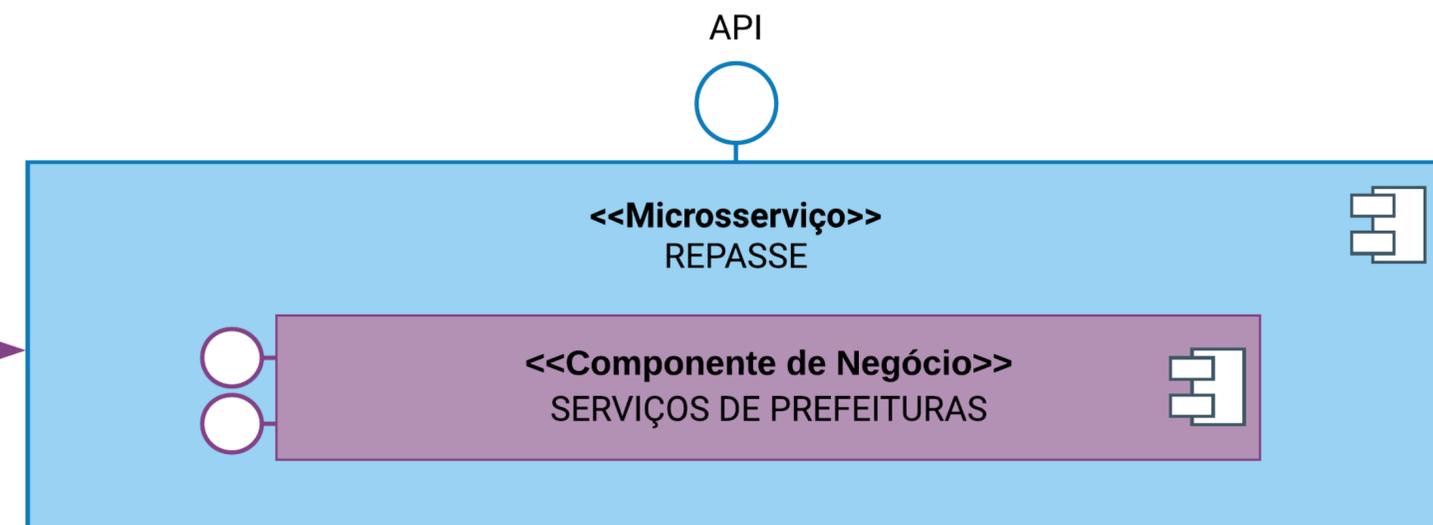
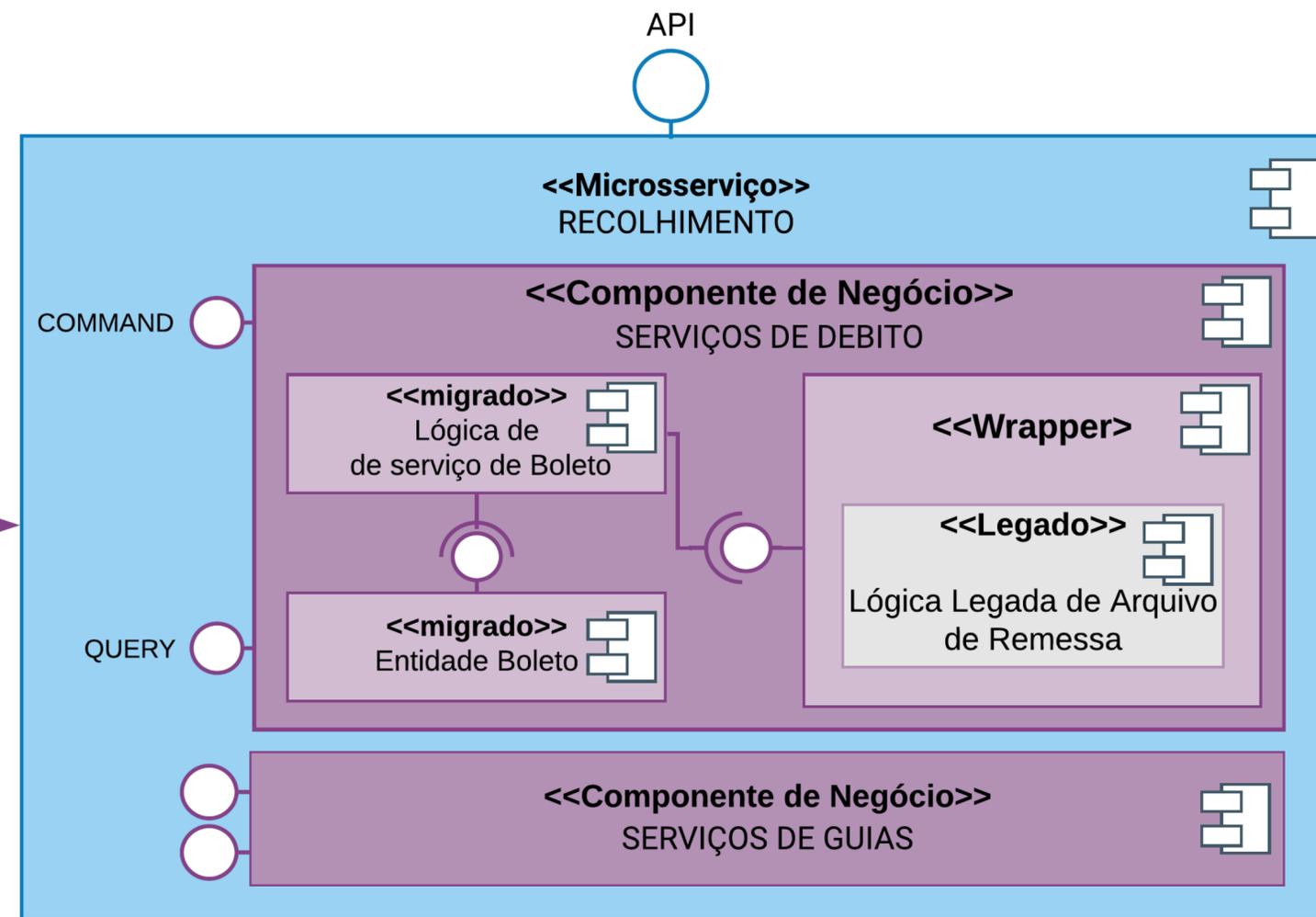
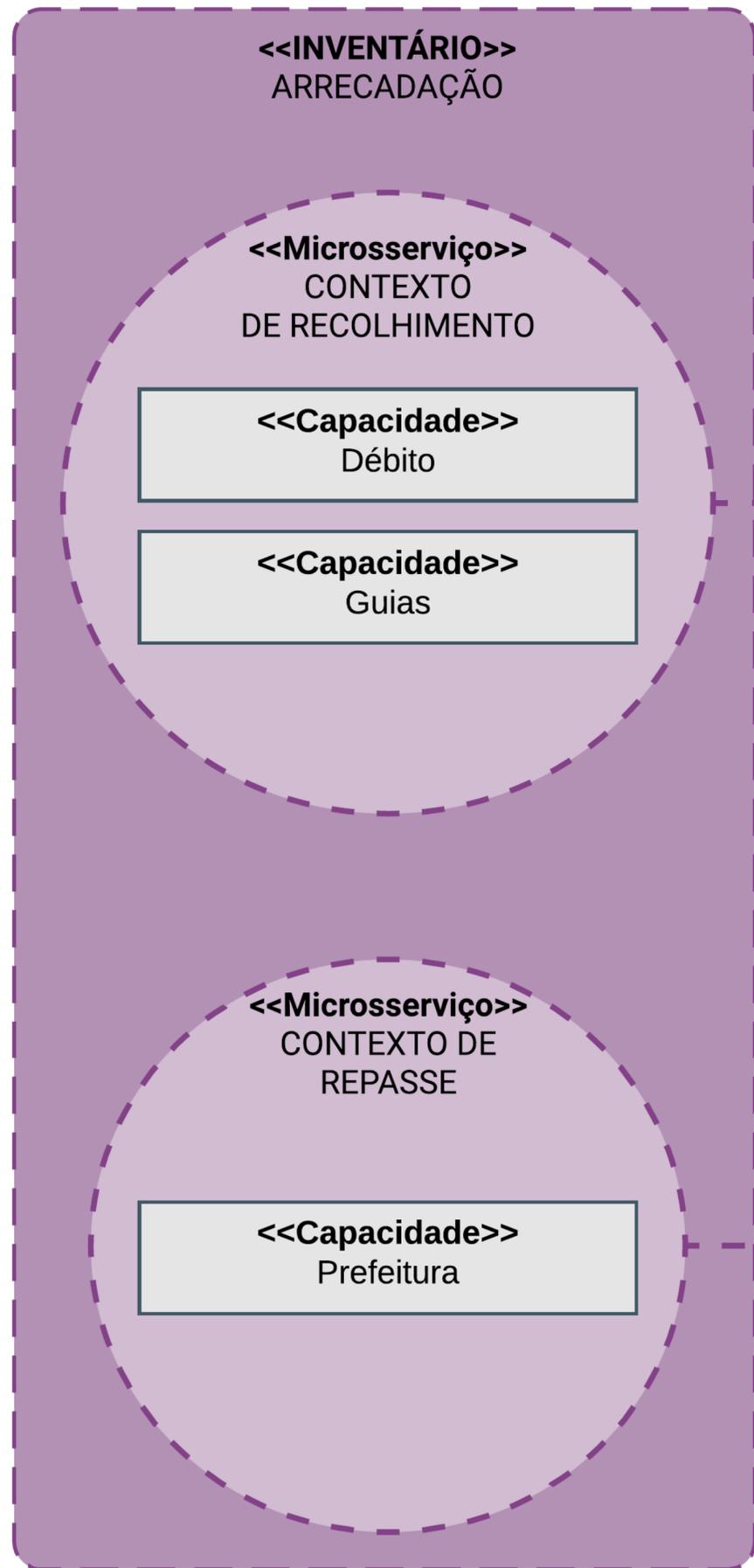


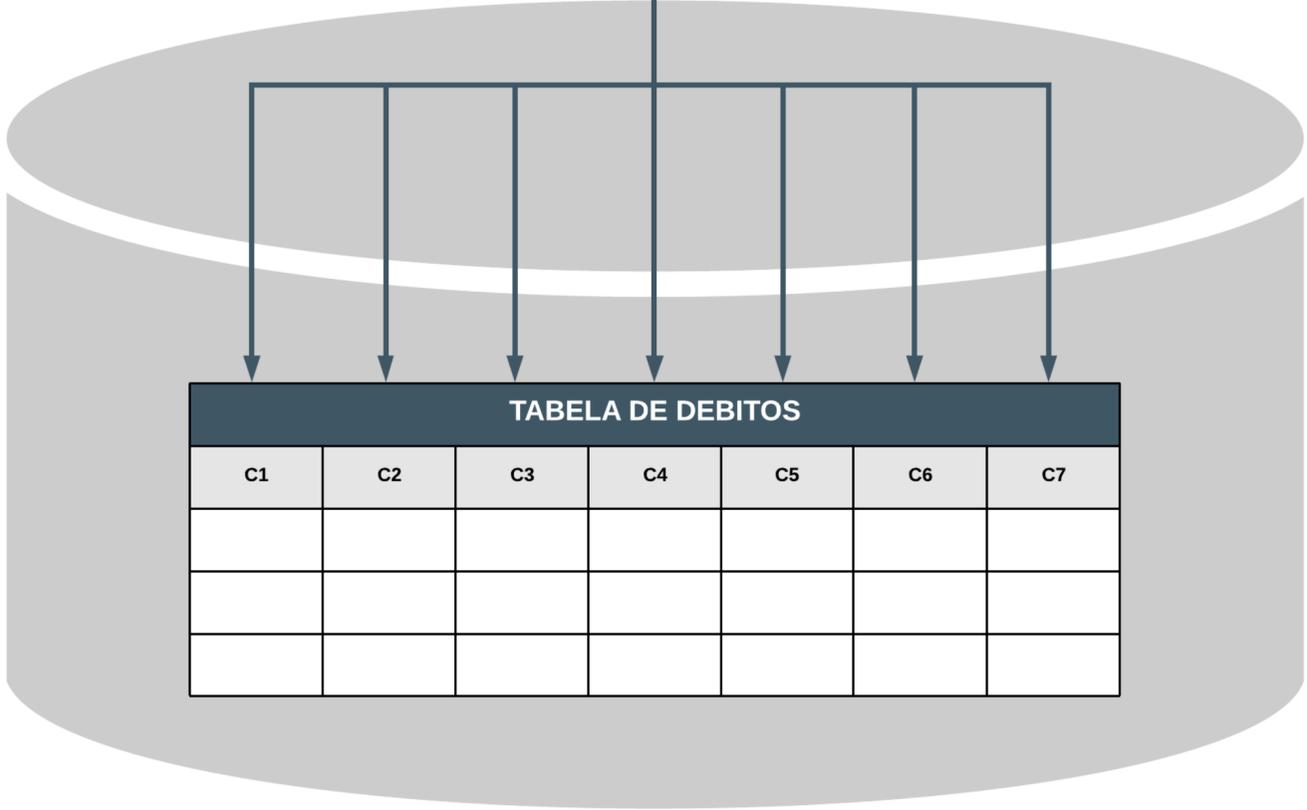
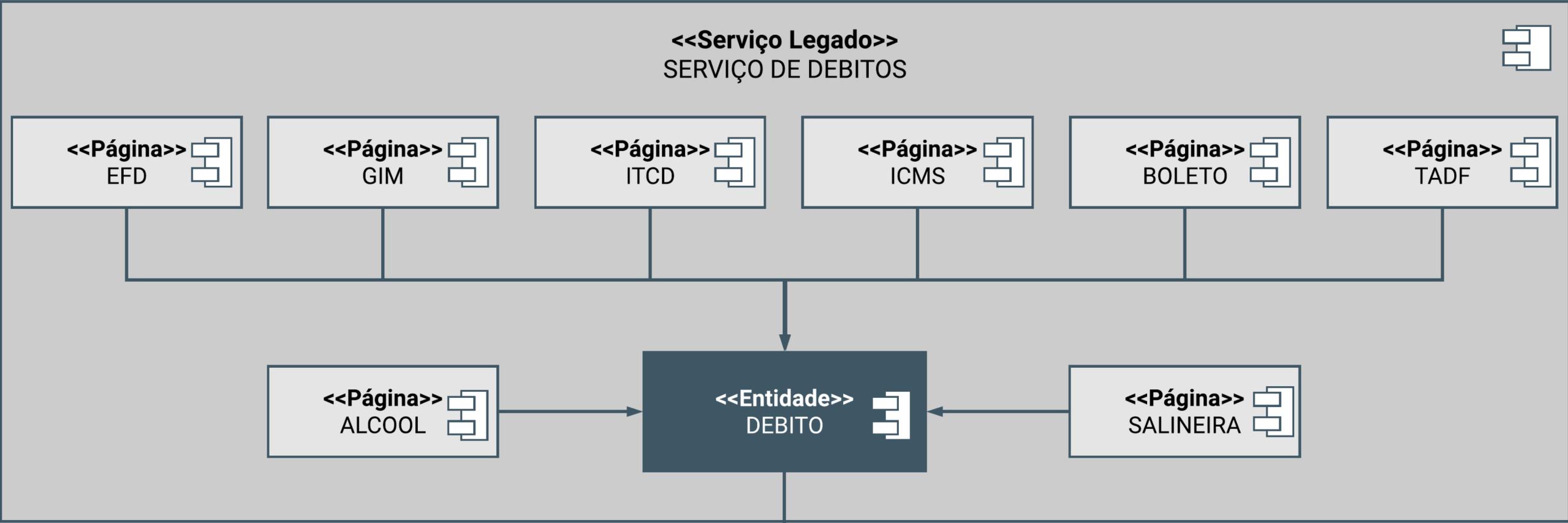
A single bounded context exposed via a web service with its own database

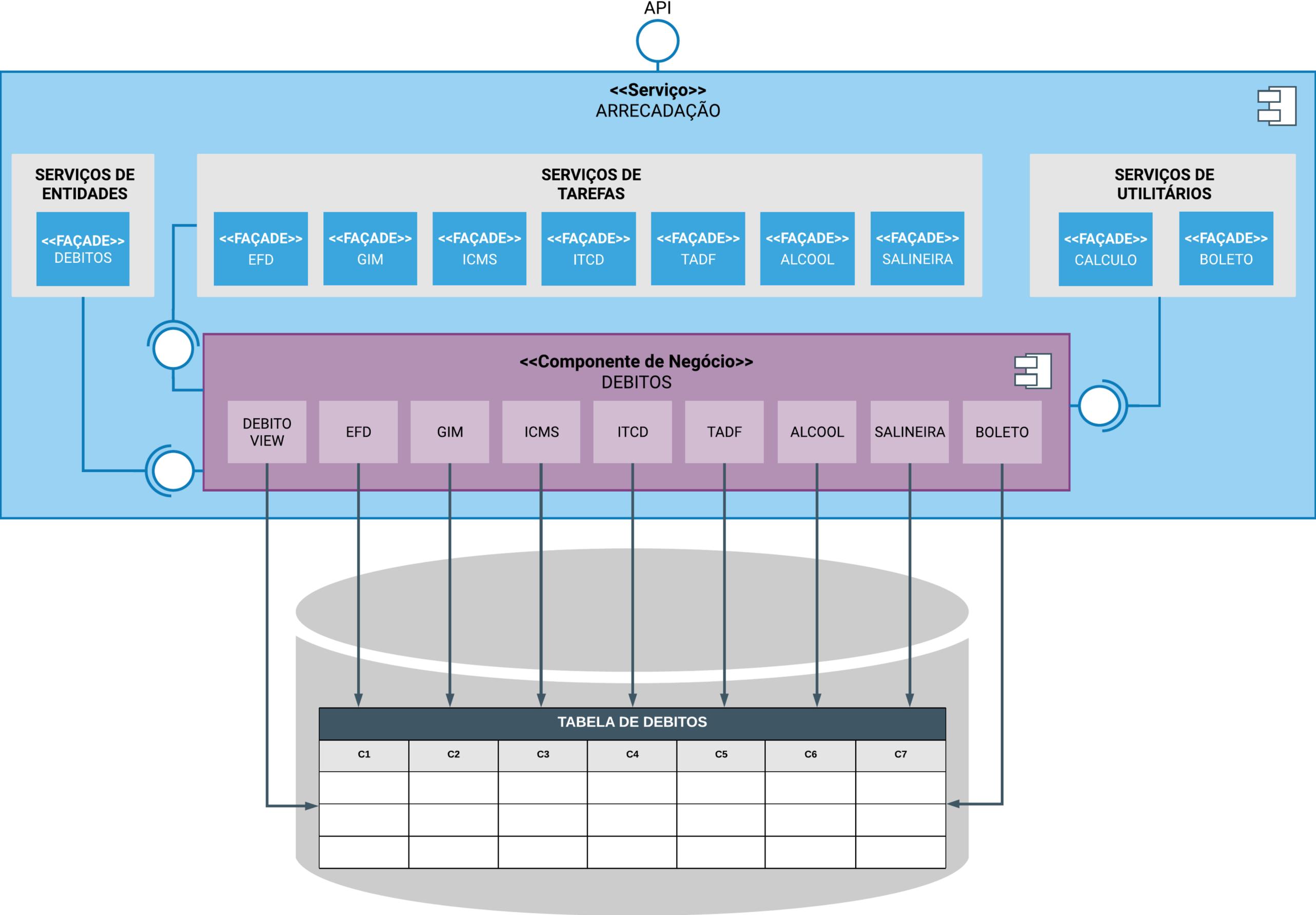


**DESIGN**

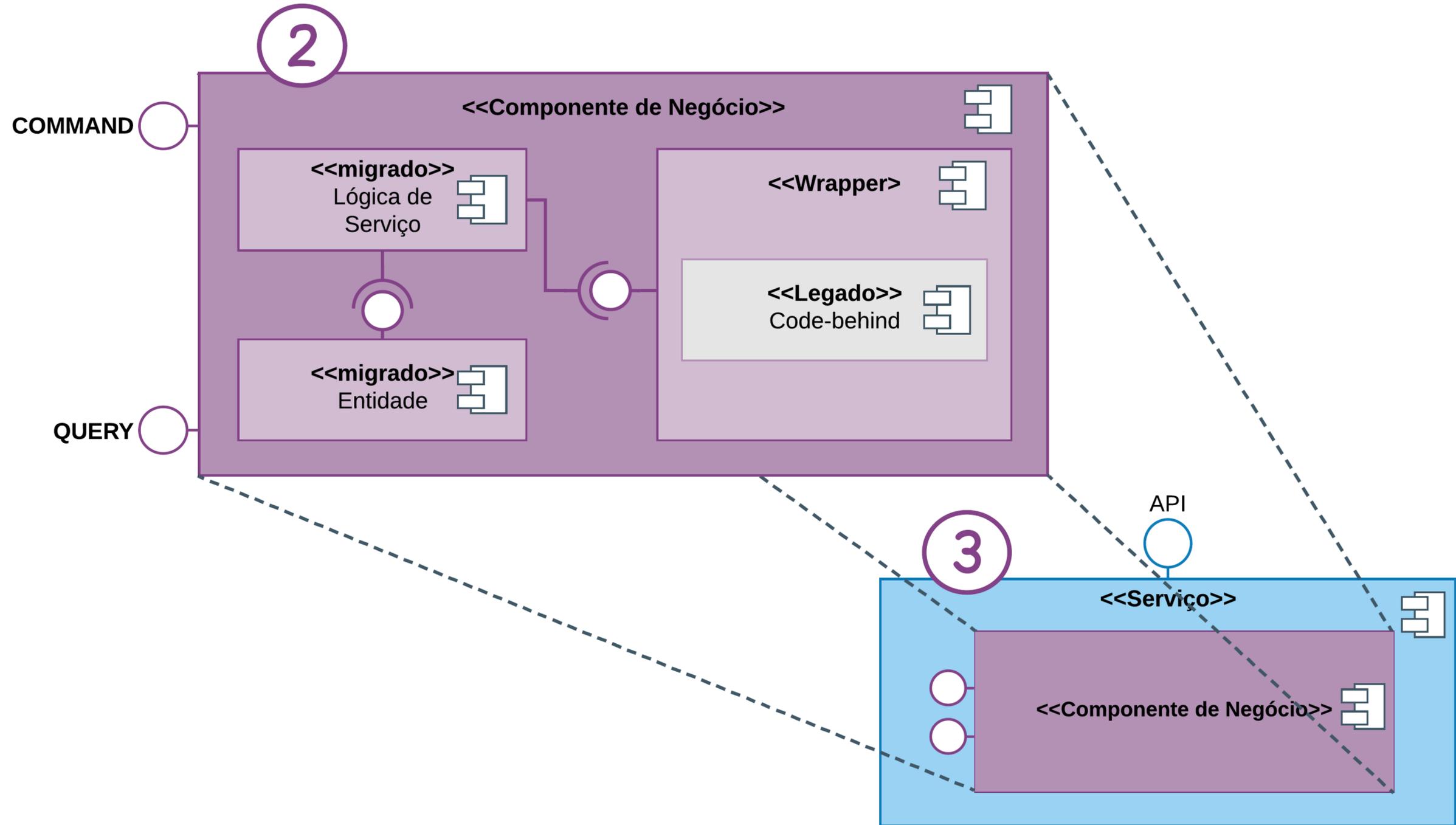








**CONSTRUÇÃO**



**<<INVENTÁRIO>>**  
ARRECADAÇÃO

**<<Microserviço>>**  
CONTEXTO  
DE RECOLHIMENTO

**<<Microserviço>>**  
CONTEXTO DE  
REPASSE

 **Solution**

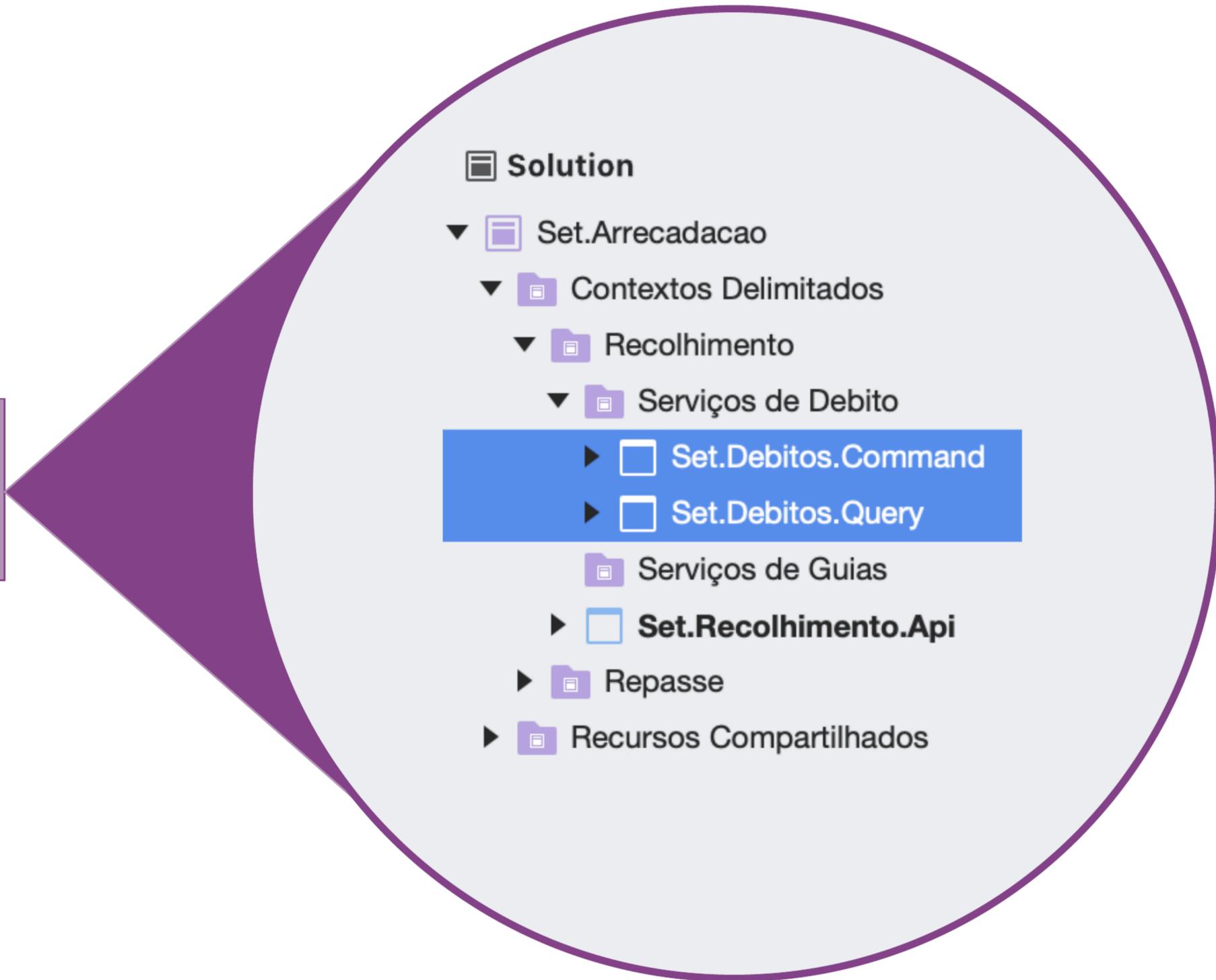
▼  Set.Arrecadacao

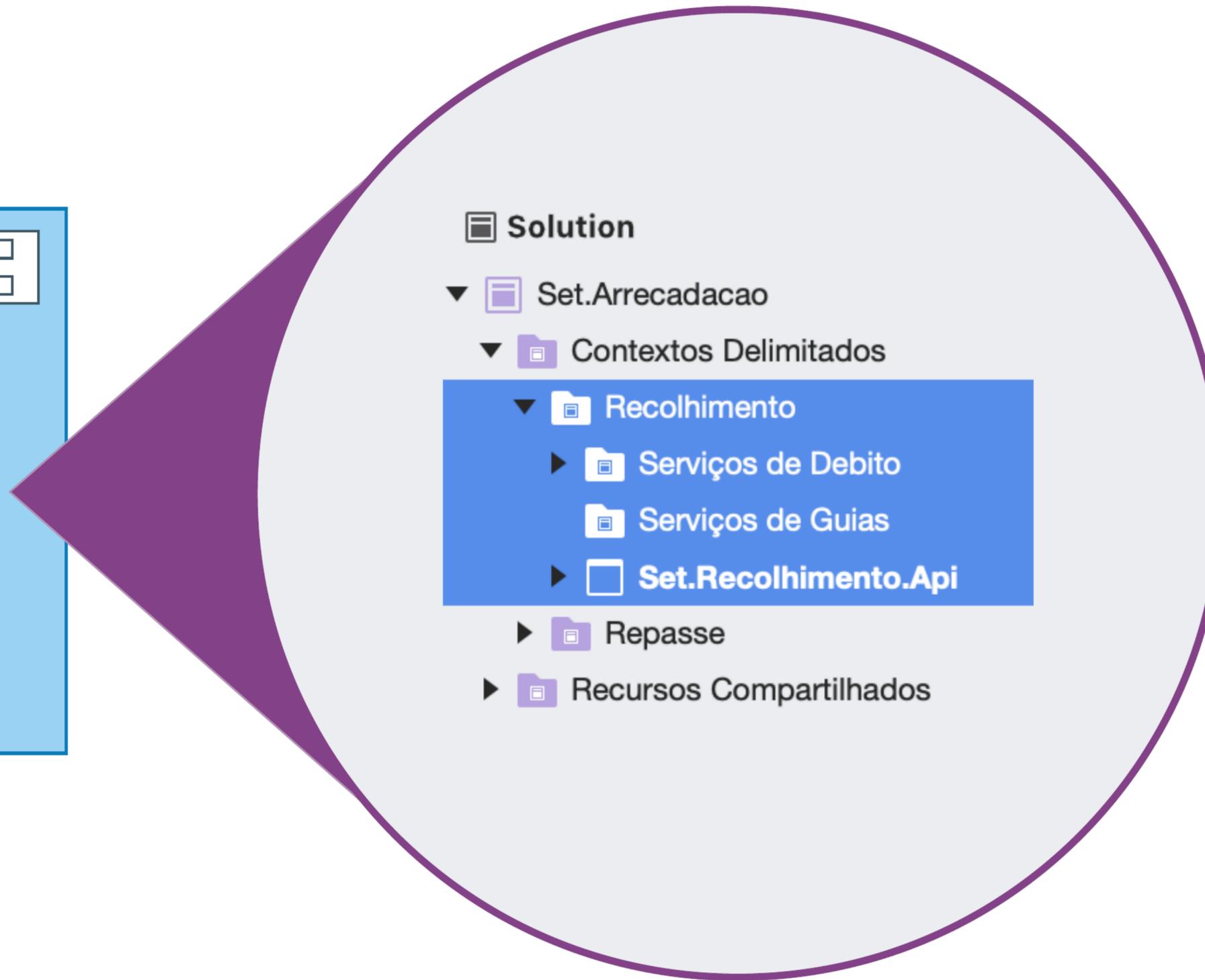
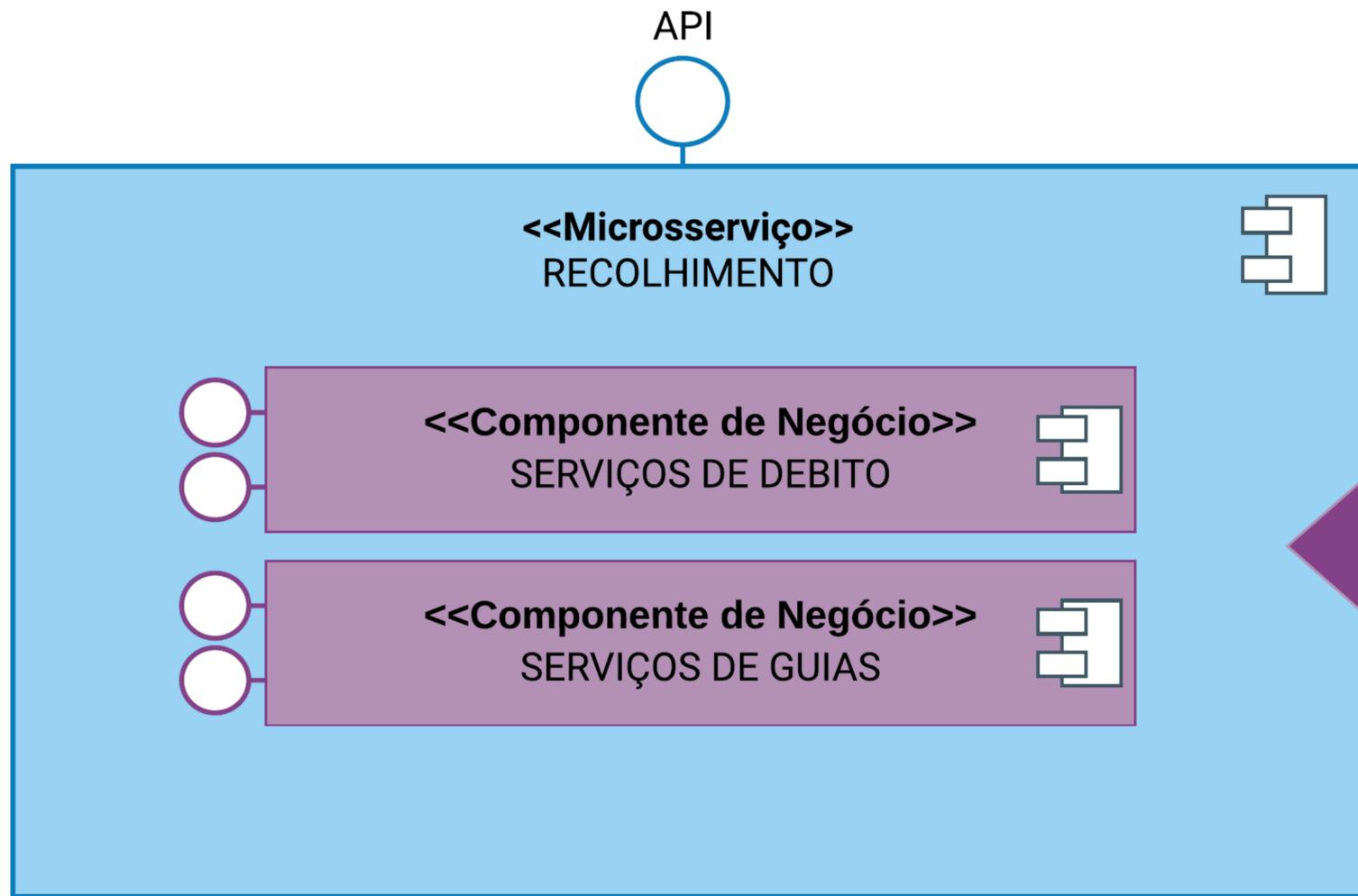
▼  Contextos Delimitados

▶  **Recolhimento**

▶  **Repasse**

▶  Recursos Compartilhados



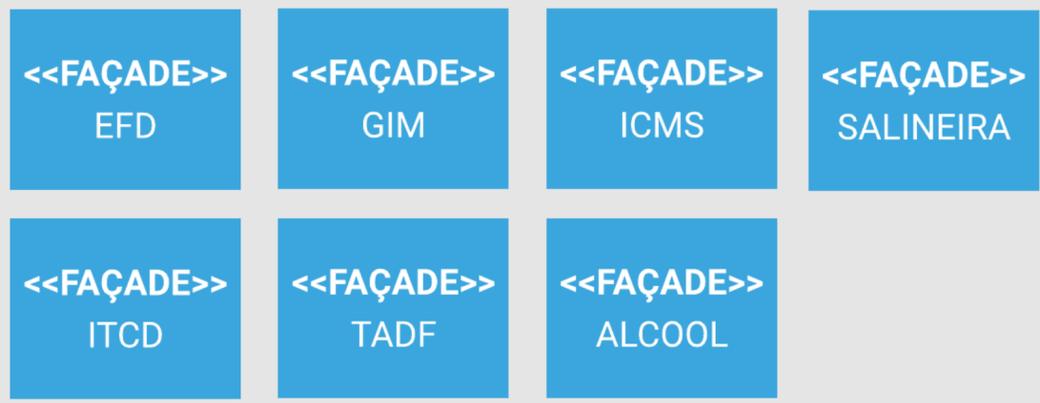




**<<Microserviço>>  
RECOLHIMENTO**



**SERVIÇOS DE  
TAREFAS**



**SERVIÇOS DE  
ENTIDADES**



**SERVIÇOS DE  
UTILITÁRIOS**



▼ **Set.Recolhimento.Api**

▶ Dependencies

▼ Debitos

▼ **Entidades**

DebitosController.cs

▼ **Tarefas**

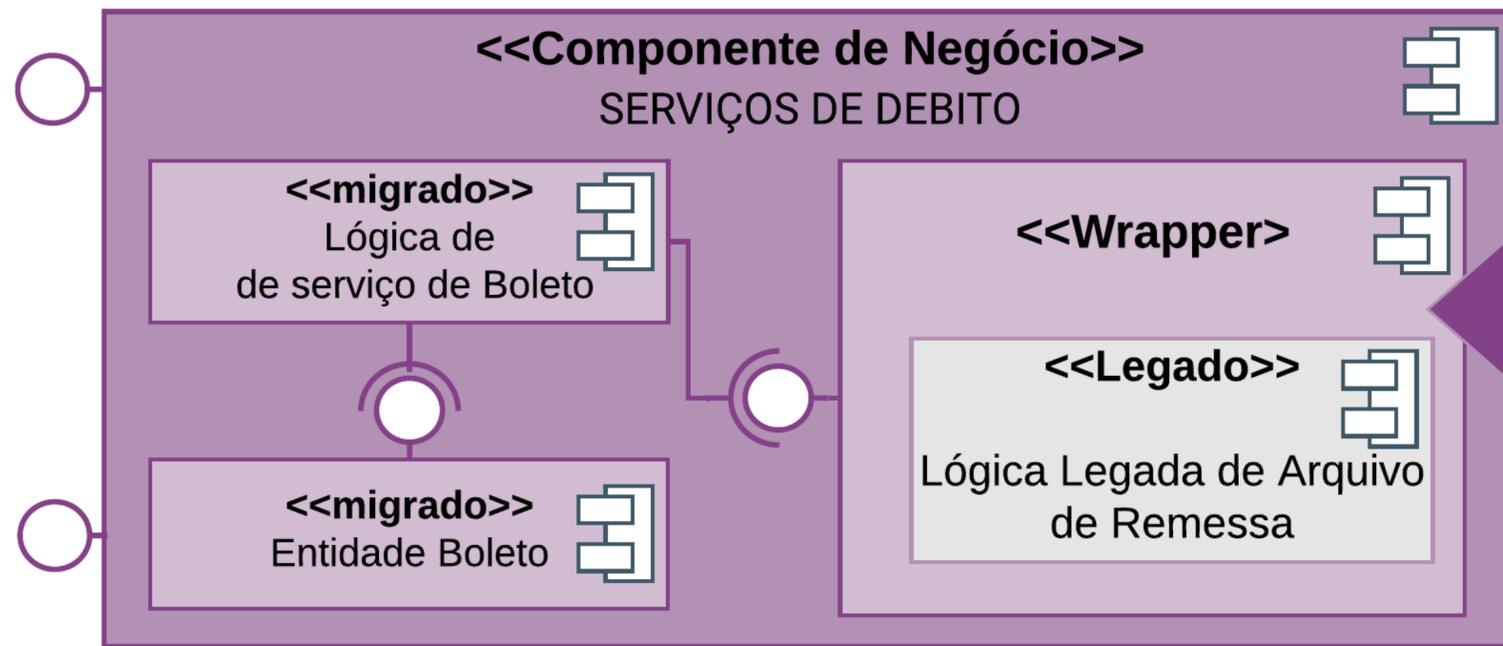
- DebitoAlcoolController.cs
- DebitoEfdController.cs
- DebitoGimController.cs
- DebitoIcmsFreteController.cs
- DebitoItcdController.cs
- DebitoSalineiraController.cs
- DebitoTadfController.cs

▼ **Utilitarios**

- BoletoController.cs
- CalculoImpostoController.cs

Program.cs

Startup.cs

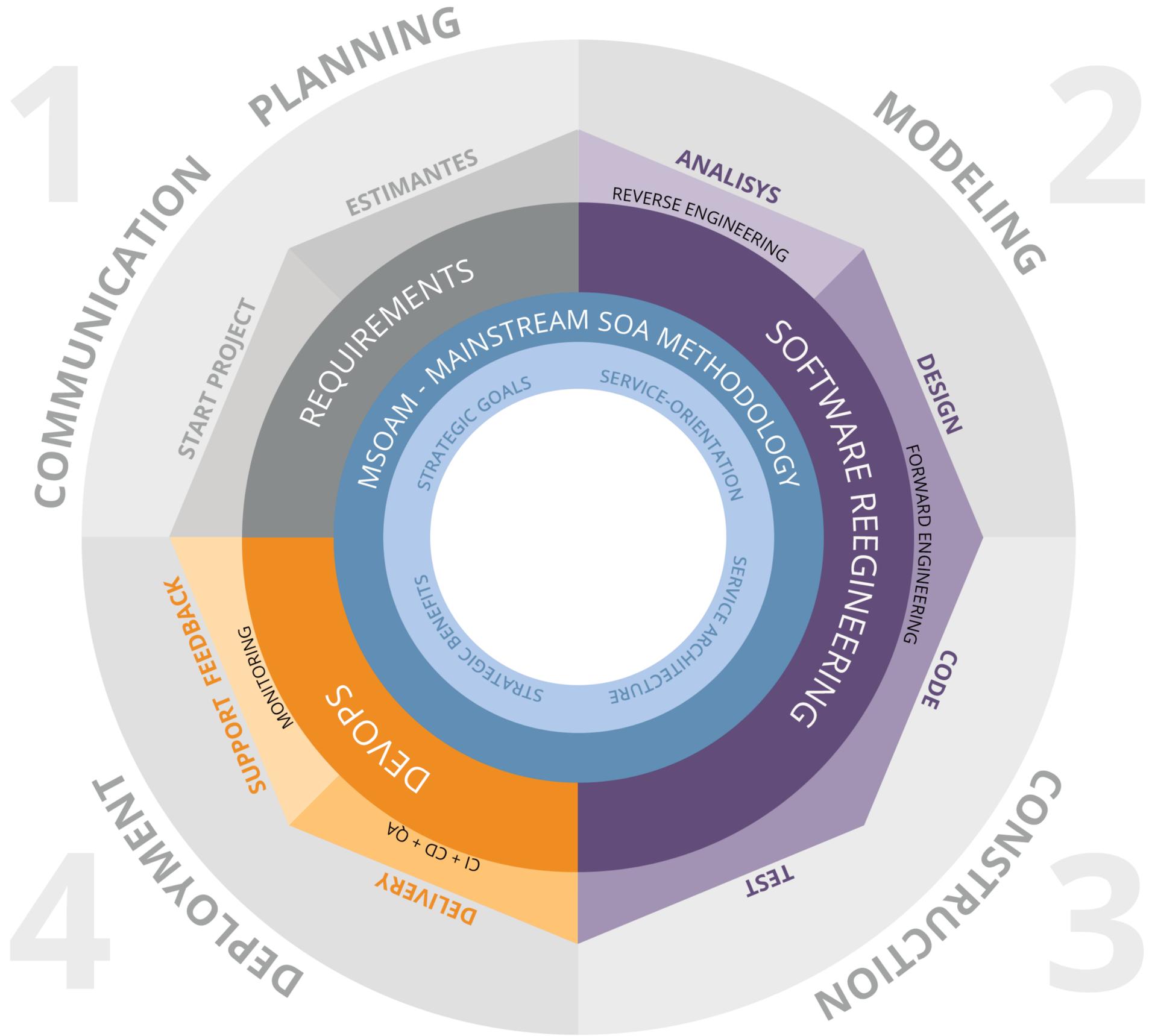


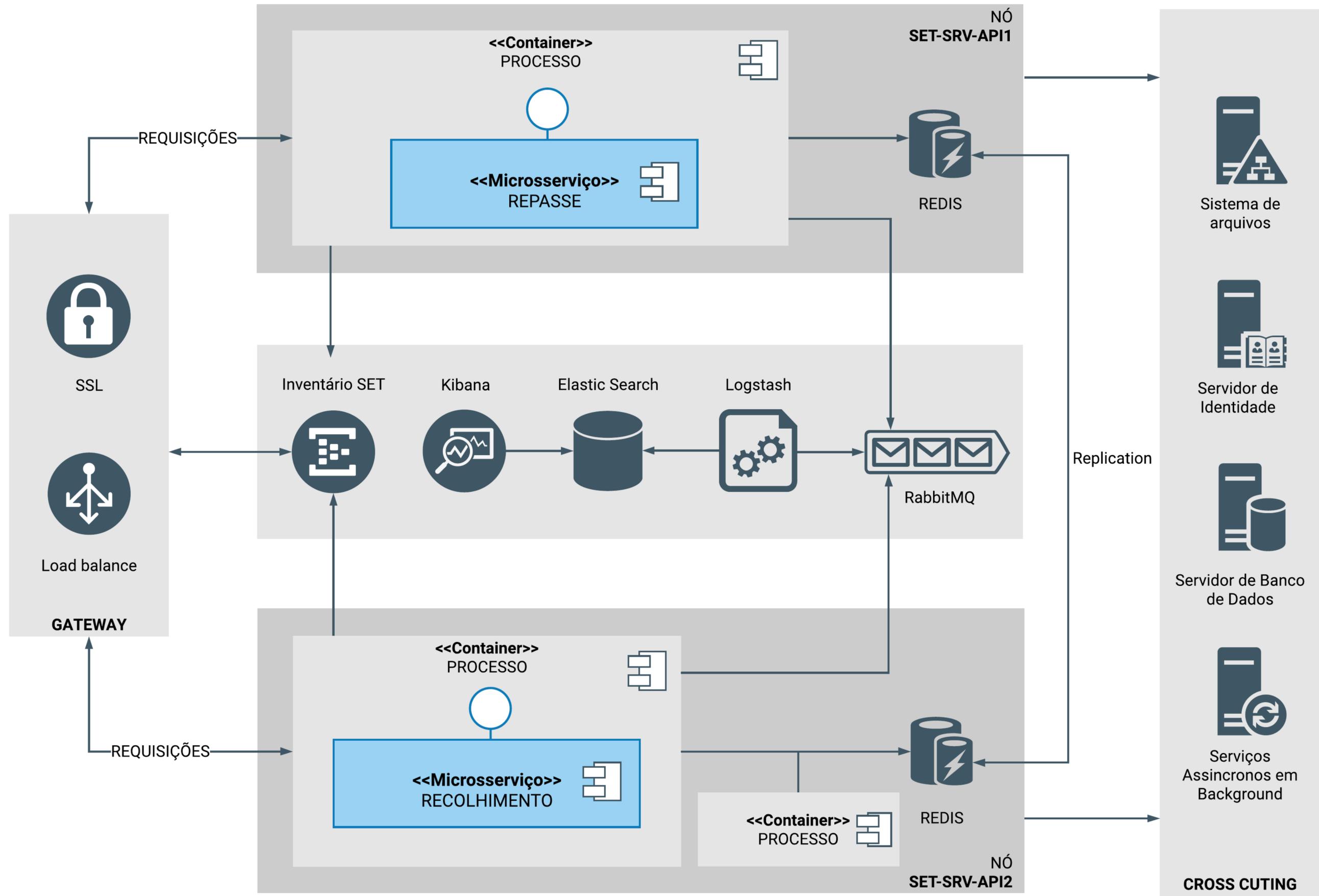
- ▼ Serviços de Debito
  - ▼ Set.Debitos.Command
    - ▶ Dependencies
    - ▶ Modelos
    - ▼ Servicos

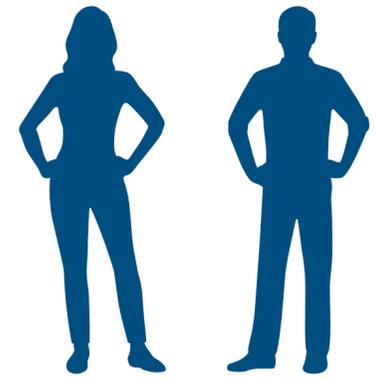
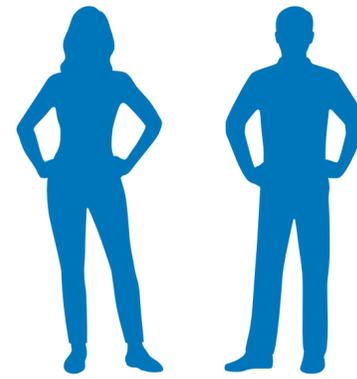
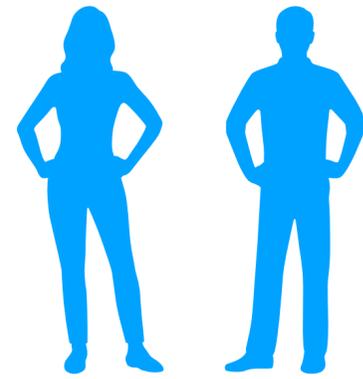
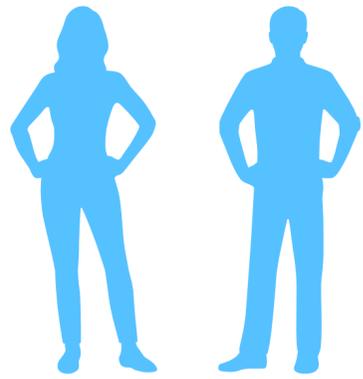
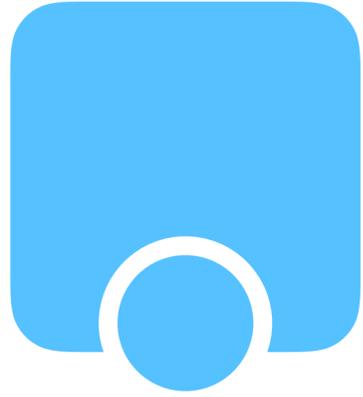
ServiçoGeracaoArquivoRemessa.cs

GeradorArquivoRemessaLegacy.cs

IGeradorArquivoRemessa.cs







Total Requisições

# 25,511

Requisições Realizadas

Total Requisição 200

# 22,561

Requisições com Sucesso

Total Requisições 4xx

# 1,721

Requisições Inválidas

Total Exceções

# 1,000

Erros de Servidor

Média de Tempo Requisição

# 168.152

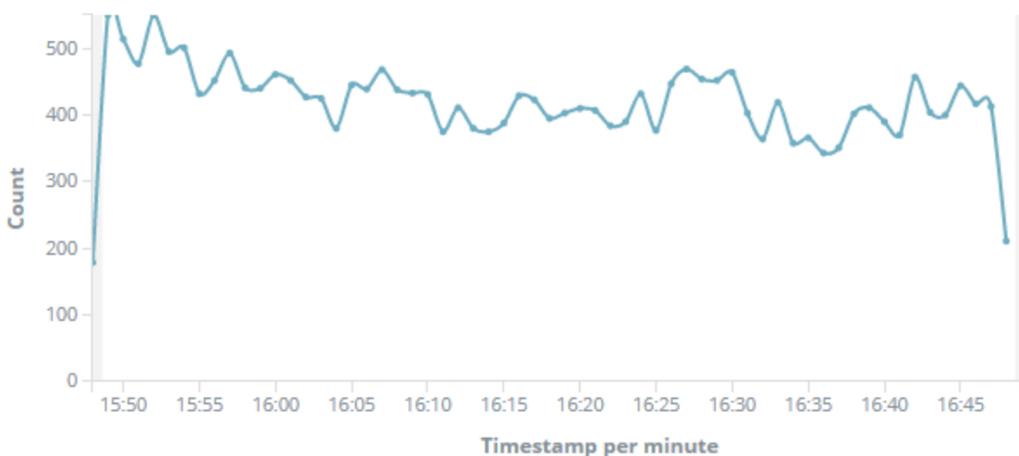
Milissegundos

Total de Usuários Autenticados

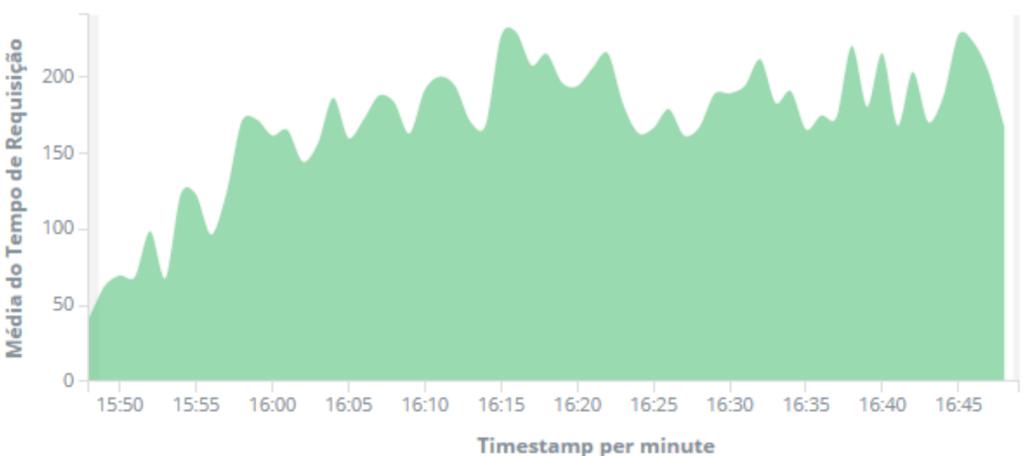
# 979

Usuários

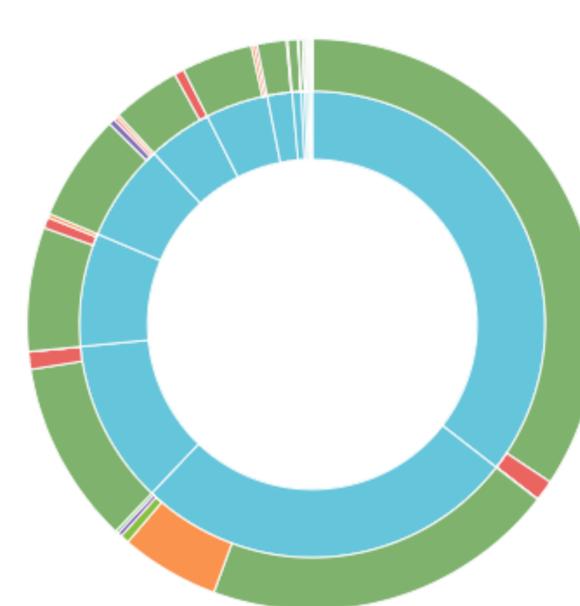
Timeline Qtd. de Requisições



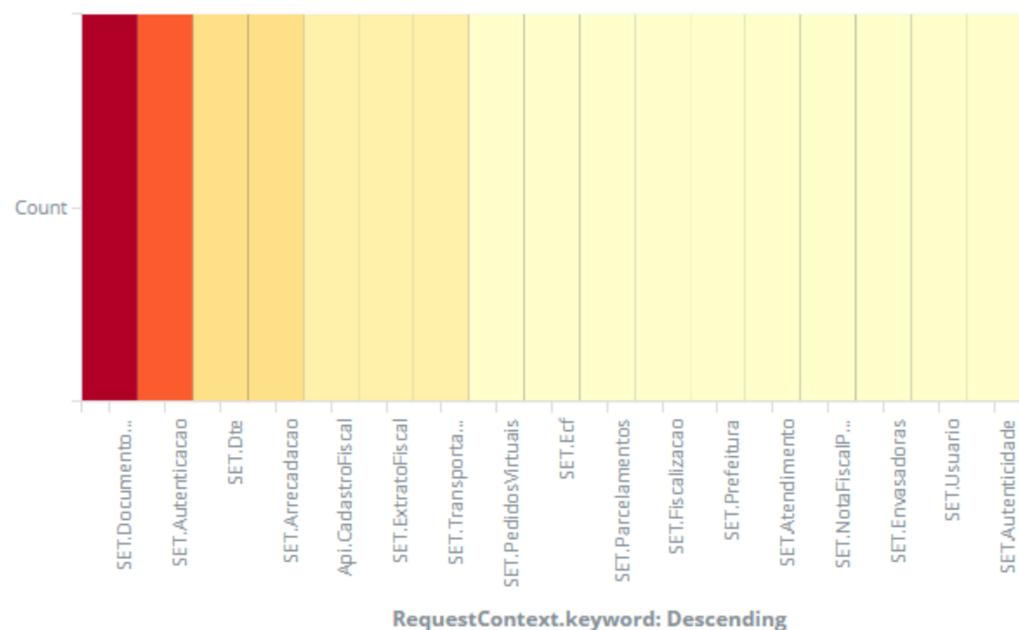
Timeline Tempo Requisição



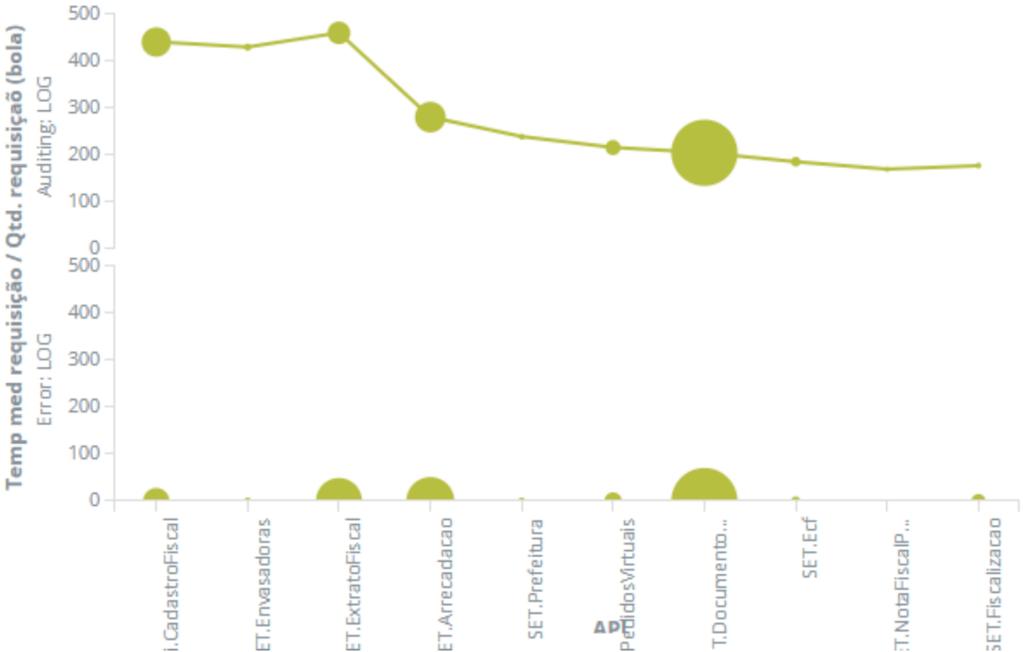
Total por tipo de Status



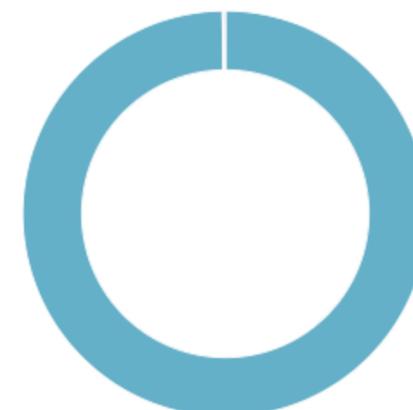
Espectro APIs x Qtd. Requisição



Tempo Médio de Requisição por API



Acesso por Aplicativo



**AVALIAÇÃO**

Tabela 3 – Dívida técnica do sistema UVT antes e depois da migração

Catégria	Código Legado			Código Migrado		
	Esforço	Custo	Correções	Esforço	Custo	Correções
<b>.NET Framework</b>						
System	6h45min	338	33	1d3h	596	36
Collection	2h10min	108	13	1d	408	49
Globalization	3d1h	1.25K	179	1d6h	742	105
Reflection	6h30min	325	39	1h50min	38	11
InteropServices	35min	29.2	7	30min	25	6
System.Threading	40min	33.3	1	-	-	-
System.Xml	3h50min	192	23	4h	200	24
<b>Project Rules</b>						
Architecture	176d	70.6K	737	8d	3.21k	135
Code Smells	51d	20.6K	273	22d	8.87k	301
Dead Code	-	-	-	6h30min	325	37
Design	4d1h	1.65K	449	4d5h	1.86k	363
Immutability	17d7h	7.17K	889	6h32min	327	53
Naming Conventions	1h25min	70.8	66	3d1h	1.29k	249
OOP Design	8d5h	3.46K	837	2d4h	1.05k	610
Source Organization	3h45min	15	188	7h	350	28
Visibility	2d4h	1.02K	656	4h22min	219	421
<b>Total</b>	<b>267d</b>	<b>107K</b>	<b>4217</b>	<b>48d</b>	<b>19.6k</b>	<b>2428</b>

Tabela 5 – Resultados da eficiência de desempenho dos serviços)

Month	#Requests	Response time (ms)	#Users
Jun/2017	197,458	222,00	939
Jul/2017	484,665	184.92	4,079
Aug/2017	652,455	199.21	5,162
Sep/2017	3,889,923	187.69	11,156
Oct/2017	4,039,758	242.07	11,347
Nov/2017	4,286,200	204.58	15,301
Dec/2017	4,447,635	203.31	14,519



**OBRIGADO**

# REFERÊNCIAS

- BAGHDADI, Y.; AL-BULUSHI, W. A guidance process to modernize legacy applications for soa. *Service Oriented Computing and Applications*, v. 9, n. 1, p. 41–58, 2015. ISSN 1863-2394. Disponível em: <<http://dx.doi.org/10.1007/s11761-013-0137-3>>. Citado na página 76.
- BASS, L.; WEBER, I.; ZHU, L. *DevOps: A Software Architect's Perspective*. [S.l.]: Addison-Wesley Professional, 2015. Citado 3 vezes nas páginas 15, 25 e 26.
- BENANY, M. M. E.; BEQQALI, O. E. Soa based e-government interoperability. In: *2015 IEEE/ACS 12th International Conference of Computer Systems and Applications (AICCSA)*. [S.l.: s.n.], 2015. p. 1–2. Citado na página 78.
- BENNETT, K. Legacy systems: Coping with success. *IEEE software*, IEEE, v. 12, n. 1, p. 19–23, 1995. Citado na página 14.
- BENNETT, K. H.; RAJLICH, V. T. Software maintenance and evolution: a roadmap. In: *ACM. Proceedings of the Conference on the Future of Software Engineering*. [S.l.], 2000. p. 73–87. Citado na página 14.
- ERL, T. *Service-oriented architecture (SOA): concepts, technology, and design*. [S.l.]: Prentice Hall, 2005. Citado 2 vezes nas páginas 16 e 26.
- ERL, T. *Soa: principles of service design*. [S.l.]: Prentice Hall Upper Saddle River, 2008. v. 1. Citado 3 vezes nas páginas 20, 21 e 26.
- ERL, T. et al. *SOA with REST: Principles, Patterns & Constraints for Building Enterprise Solutions with REST*. 1st. ed. Upper Saddle River, NJ, USA: Prentice Hall Press, 2012. ISBN 0137012519, 9780137012510. Citado 5 vezes nas páginas 26, 44, 49, 62 e 79.
- ERL, T. et al. *Next Generation SOA: A Concise Introduction to Service Technology & #38; Service-Orientation*. 1st. ed. Upper Saddle River, NJ, USA: Prentice Hall Press, 2014. ISBN 0133859045, 9780133859041. Citado 3 vezes nas páginas 21, 26 e 50.
- ERL, T.; MERSON, P.; STOFFERS, R. *Service-Oriented Architecture: Analysis and Design for Services and Microservices*. [S.l.]: Prentice Hall, 2017. Citado 9 vezes nas páginas 9, 14, 16, 20, 21, 22, 23, 26 e 53.
- EVANS. *Domain-Driven Design: Tacking Complexity In the Heart of Software*. Boston, MA, USA: Addison-Wesley Longman Publishing Co., Inc., 2003. ISBN 0321125215. Citado 7 vezes nas páginas 25, 36, 43, 45, 46, 47 e 50.
- FOWLER, S. J. *Production-Ready Microservices: Building Standardized Systems Across an Engineering Organization*. 1st. ed. [S.l.]: O'Reilly Media, Inc., 2016. ISBN 1491965975, 9781491965979. Citado 2 vezes nas páginas 25 e 26.
- GRUBB, P.; TAKANG, A. A. *Software maintenance: concepts and practice*. [S.l.]: World Scientific, 2003. Citado 2 vezes nas páginas 14 e 24.
- JAMSHIDI, P. et al. Microservices: The journey so far and challenges ahead. *IEEE Software*, v. 35, n. 3, p. 24–35, May 2018. ISSN 0740-7459. Citado 3 vezes nas páginas 15, 25 e 79.
- LEON, A. F. *Don't believe the hype! SOA AND MSA are not the same*. 2017. Website. Disponível em: <[https://www.academia.edu/34828240/Dont\\_believe\\_the\\_hype\\_SOA\\_AND\\_MSA\\_are\\_not\\_the\\_same](https://www.academia.edu/34828240/Dont_believe_the_hype_SOA_AND_MSA_are_not_the_same)>. Citado 3 vezes nas páginas 14, 15 e 79.
- LEWIS, G.; SMITH, D.; KONTOGIANNIS, K. *A Research Agenda for Service-Oriented Architecture (SOA): Maintenance and Evolution of Service-Oriented Systems*. Pittsburgh, PA, 2010. Disponível em: <<http://resources.sei.cmu.edu/library/asset-view.cfm?AssetID=9285>>. Citado 2 vezes nas páginas 76 e 80.
- MILLETT, S. *Patterns, principles and practices of domain-driven design*. [S.l.]: John Wiley & Sons, 2015. Citado 2 vezes nas páginas 41 e 47.
- NEWMAN, S. *Building Microservices*. 1st. ed. [S.l.]: O'Reilly Media, Inc., 2015. ISBN 1491950358, 9781491950357. Citado 3 vezes nas páginas 15, 25 e 26.
- PARNAS, D. L. Software aging. In: *IEEE. Software Engineering, 1994. Proceedings. ICSE-16., 16th International Conference on*. [S.l.], 1994. p. 279–287. Citado na página 14.
- PRESSMAN, R. S.; MAXIM, B. A. *Software Engineering: A Practitioner's Approach, 8ª Edition*. [S.l.]: McGraw Hill, 2016. Citado 4 vezes nas páginas 23, 24, 37 e 39.
- REDDY, V. K. et al. Evaluating legacy assets in the context of migration to soa. *Software Quality Journal*, v. 17, n. 1, p. 51–63, Mar 2009. Disponível em: <<https://doi.org/10.1007/s11219-008-9055-6>>. Citado na página 76.
- RICHARDS, M. *Microservices vs. service-oriented architecture*. [S.l.]: O'Reilly Media, 2015. Citado 2 vezes nas páginas 15 e 26.
- SANTIKA, H.; SUHARDI; YUSTIANTO, P. Engineering local government financial service system under good governance principles: Case study: Cimahi government city. In: *2017 5th International Conference on Information and Communication Technology (ICoICT)*. [S.l.: s.n.], 2017. p. 1–6. Citado na página 79.
- SHAHIN, M.; BABAR, M. A.; ZHU, L. Continuous integration, delivery and deployment: A systematic review on approaches, tools, challenges and practices. *IEEE Access*, v. 5, p. 3909–3943, 2017. ISSN 2169-3536. Citado 4 vezes nas páginas 9, 16, 25 e 26.
- SHEIKH, M. A. A.; ABOALSAMH, H. A.; ALBARRAK, A. Migration of legacy applications and services to service-oriented architecture (soa). In: *The 2011 International Conference and Workshop on Current Trends in Information Technology (CTIT 11)*. [S.l.: s.n.], 2011. p. 137–142. ISSN 2377-5327. Citado na página 76.
- TRIPATHY, P.; NAIK, K. *Software evolution and maintenance*. [S.l.]: John Wiley & Sons, 2014. Citado na página 24.
- VERNON, V. *Implementing Domain-Driven Design*. Pearson Education, 2013. ISBN 9780133039887. Disponível em: <<https://books.google.com.br/books?id=X7DpD5g3VP8C>>. Citado na página 36.