

# Consulta Pública para definição de Arquitetura em Nuvem para Projetos de Dados

Rio de Janeiro - Dez/2021



# Agenda

- Objetivo
- Apresentação do Data Lake
  - **O que** queremos resolver
  - **Como** estamos resolvendo
  - **Com o que** estamos resolvendo
- Aprofundamento
  - **Coordenação** de componentes
  - **Casos** de uso
  - **Próximos** passos
  - **Perguntas**



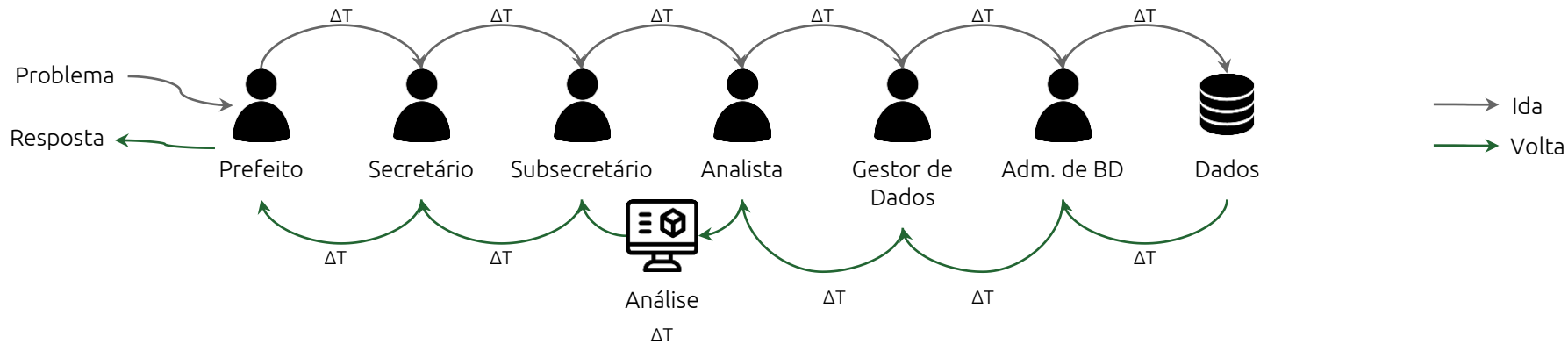
# Objetivo

- A presente Consulta Pública tem como propósito a definição técnica de uma arquitetura em nuvem para projetos de dados.
- Este documento, composto da descrição de uma arquitetura tecnologicamente agnóstica e de um anexo de requisitos, apresenta a visão do problema que se espera resolver e para o qual interessados podem apresentar suas soluções para apreciação interna.
- As propostas de soluções deverão ser preenchidas a partir de uma cópia [dessa planilha](#).
- A partir das informações apresentadas a Prefeitura do Rio realizará uma análise técnica para definição dos elementos que sustentarão a arquitetura estratégia para projetos de dados.

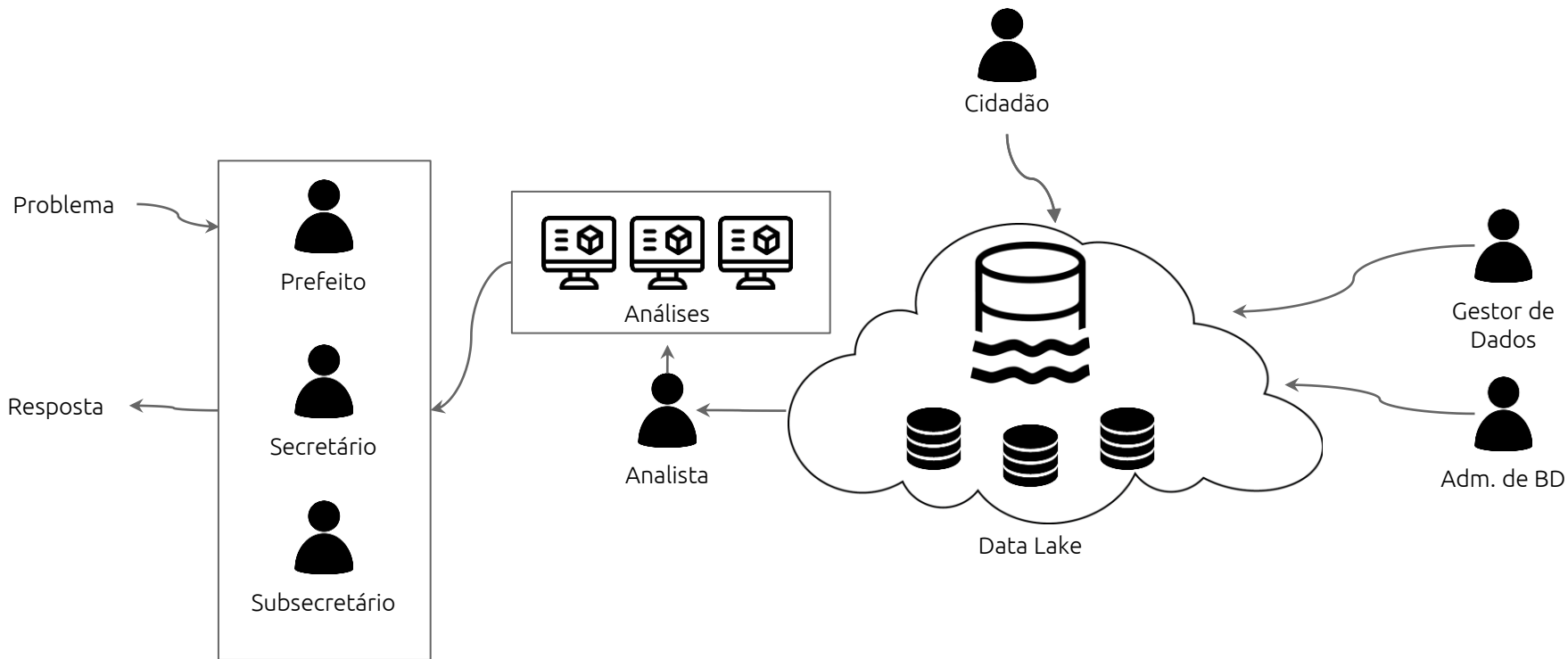


# 0 que queremos resolver

- Demora para acessar dados
- Dificuldade em localizar dados
- Dados inconsistentes
- Tecnologias desatadas: difícil de trabalhar com diferentes conjuntos de dados
- Baixa escalabilidade: escala verticalmente



# Como estamos resolvendo



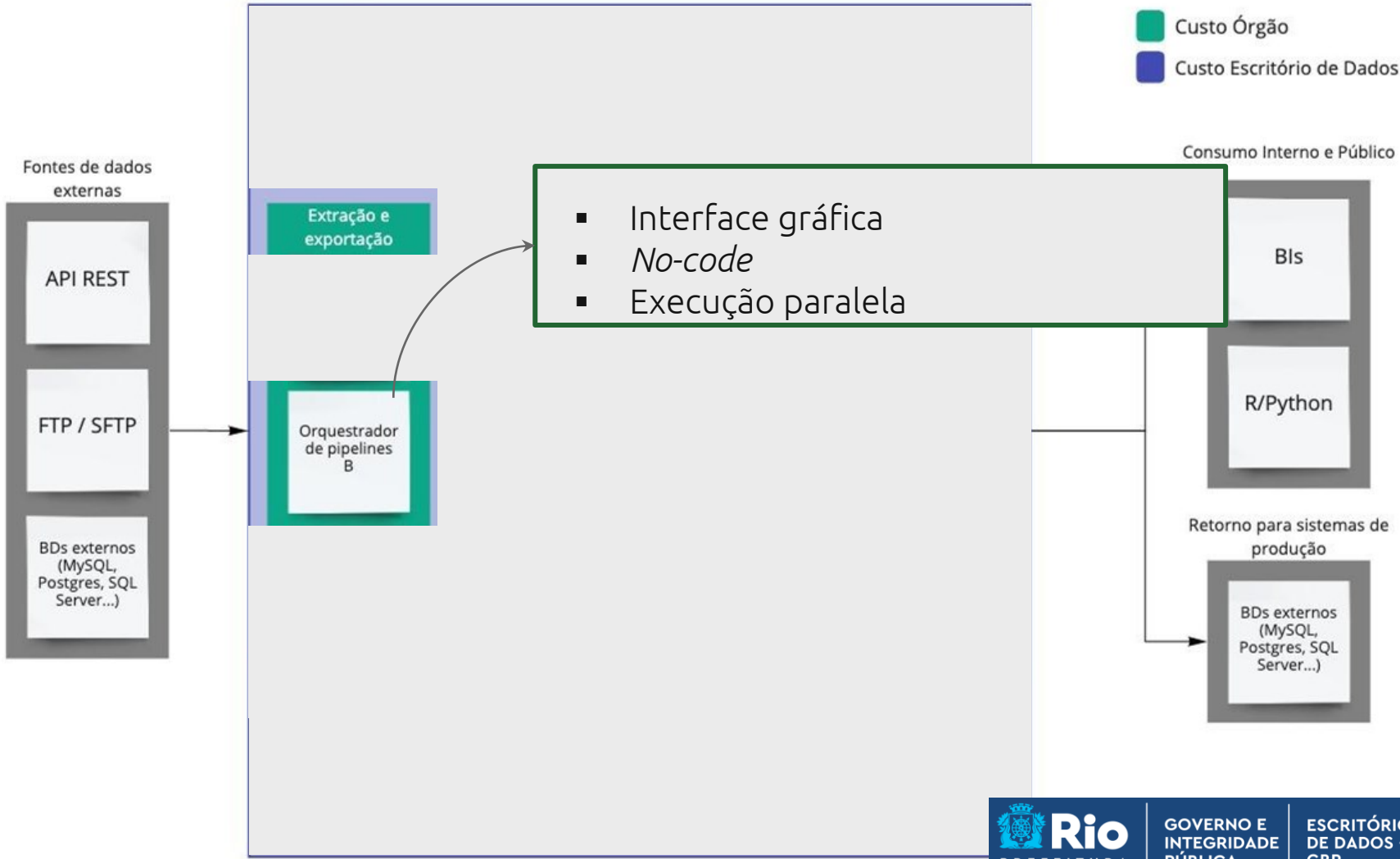
# Como estamos resolvendo

- **Custo descentralizado** e dimensionado em função do uso
- **Dados centralizados** e com uma única interface de acesso
- **Padronização, harmonização e qualidade dos dados**
- **Unidade de tecnologias** utilizadas para extração e processamento de dados
- **Padronização de estilos de código**
- **Gerenciamento de acesso aos dados** com possibilidade de disponibilizar publicamente
- **Capacidade de desacoplamento** da macroestrutura, caso necessário
- **Infraestrutura modular, escalável e de fácil manutenção**

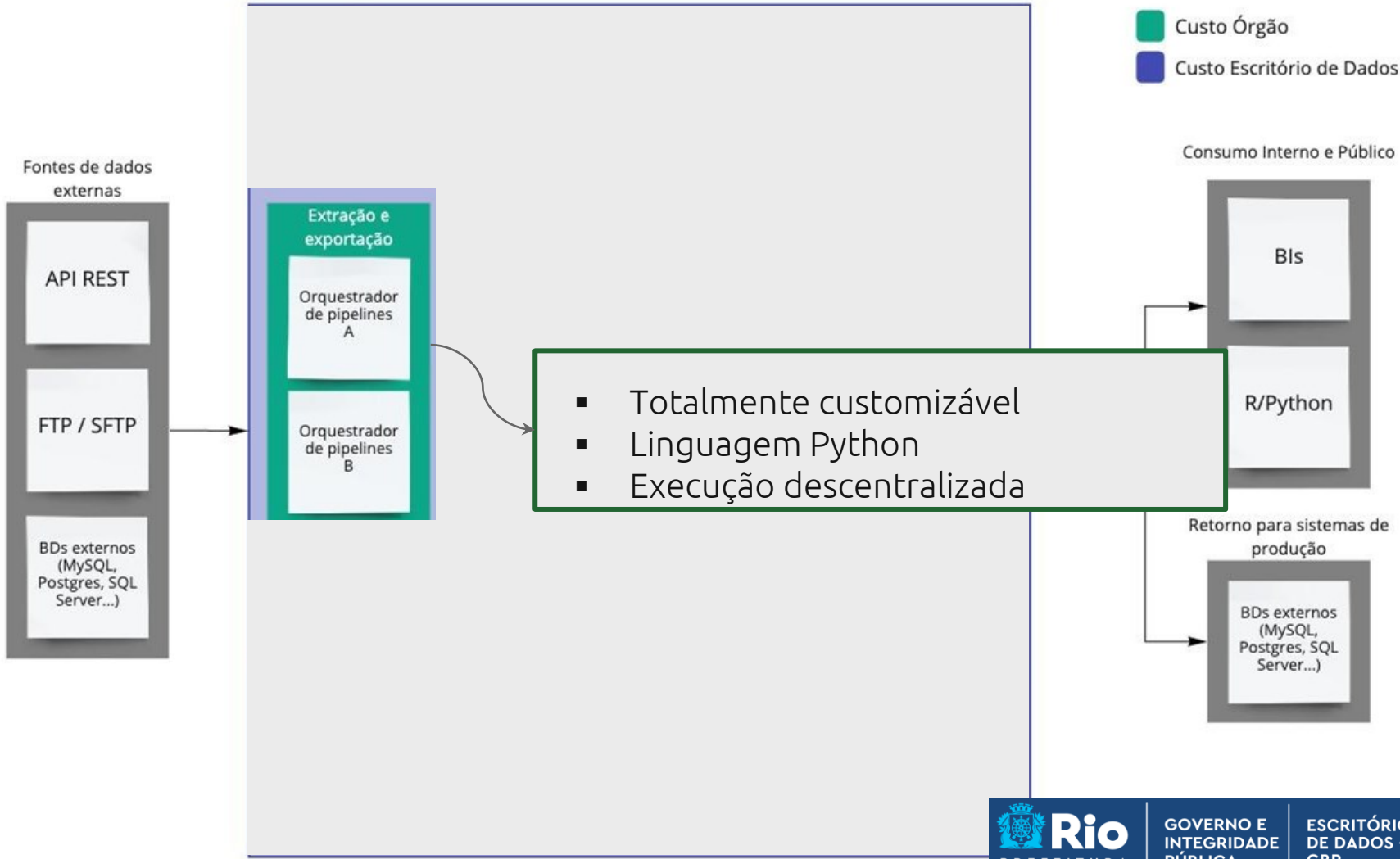


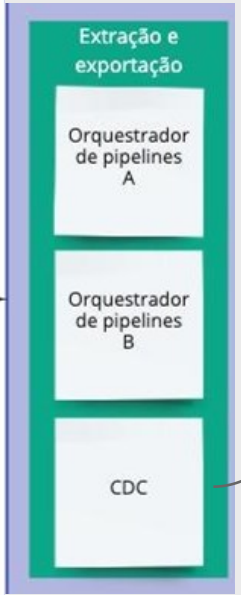
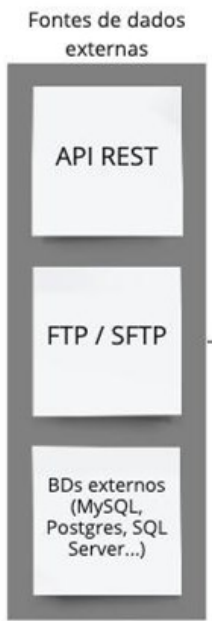
- Custo Órgão
- Custo Escritório de Dados





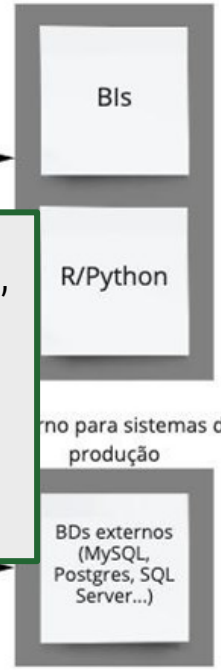


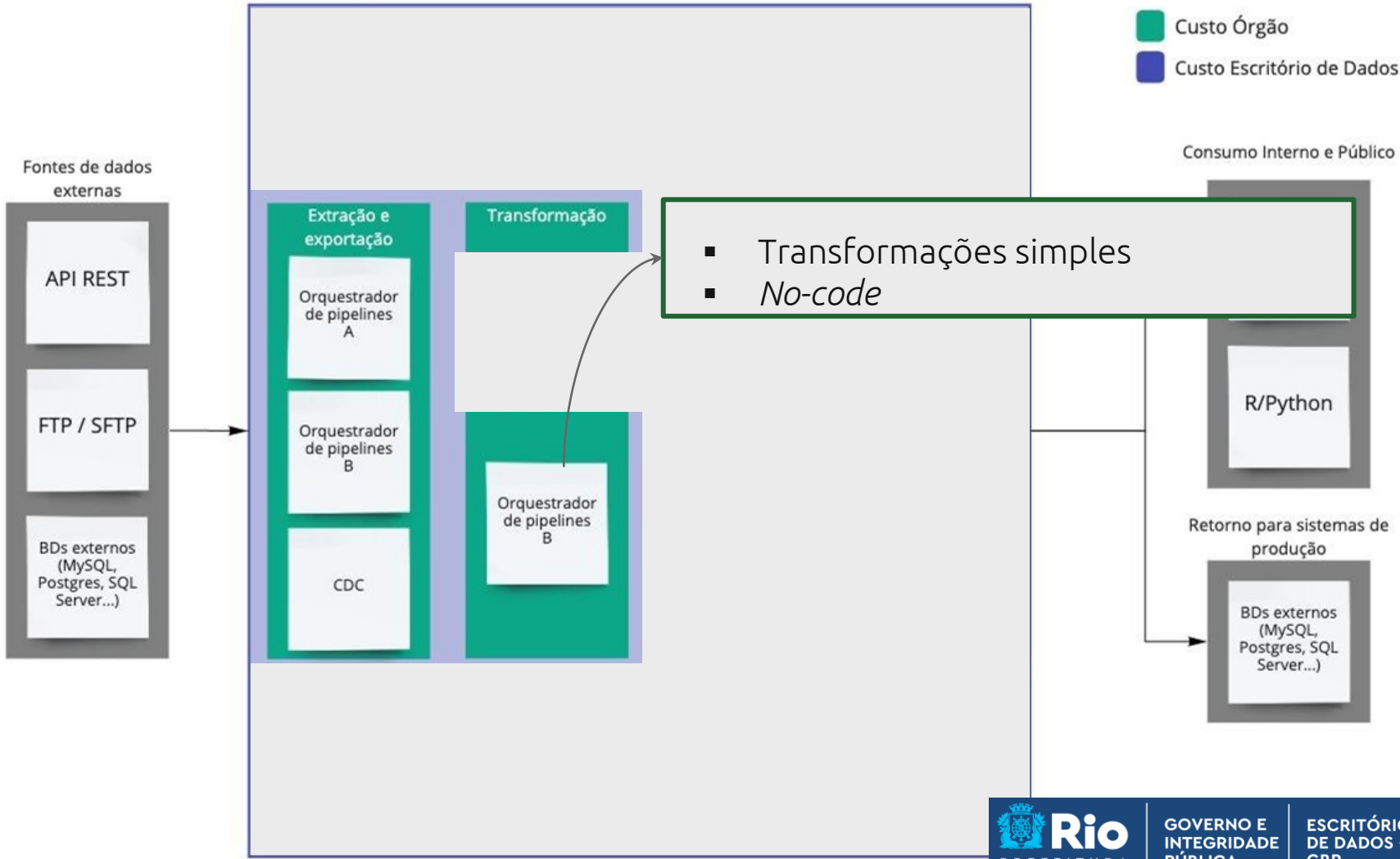


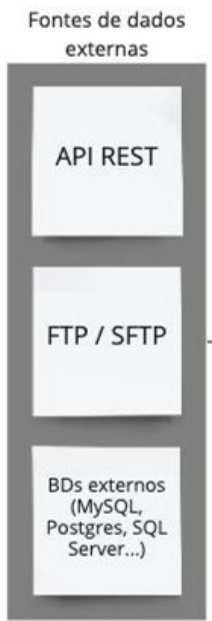


- Diversas fontes (mínimo PostgreSQL, SQL Server, MySQL)
- Deve integrar com Armazém de Dados
- Não deve submeter consultas ao SGBD

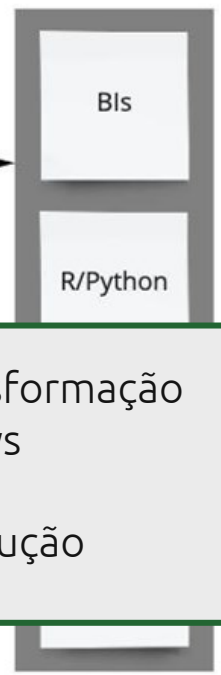
Consumo Interno e Público





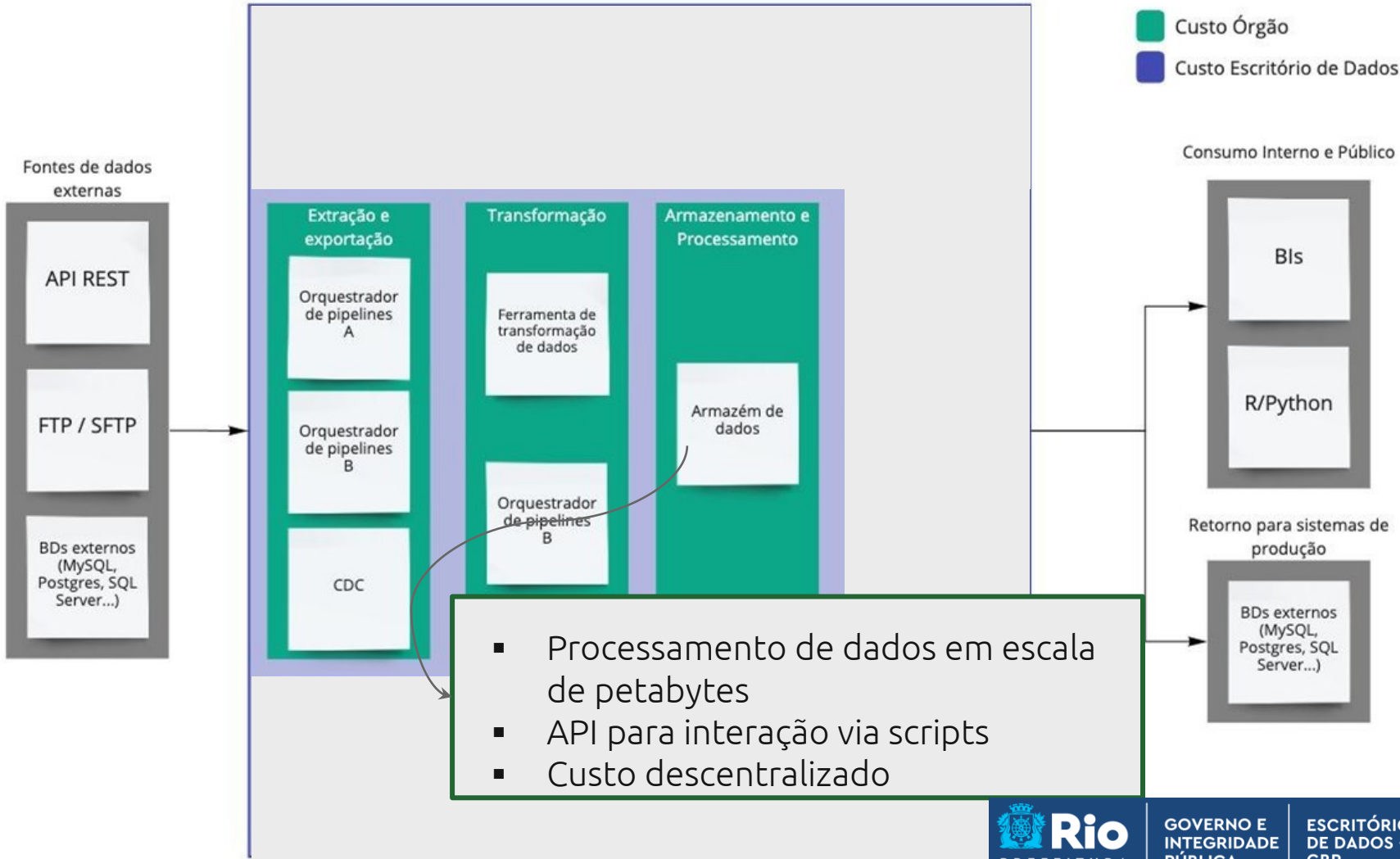


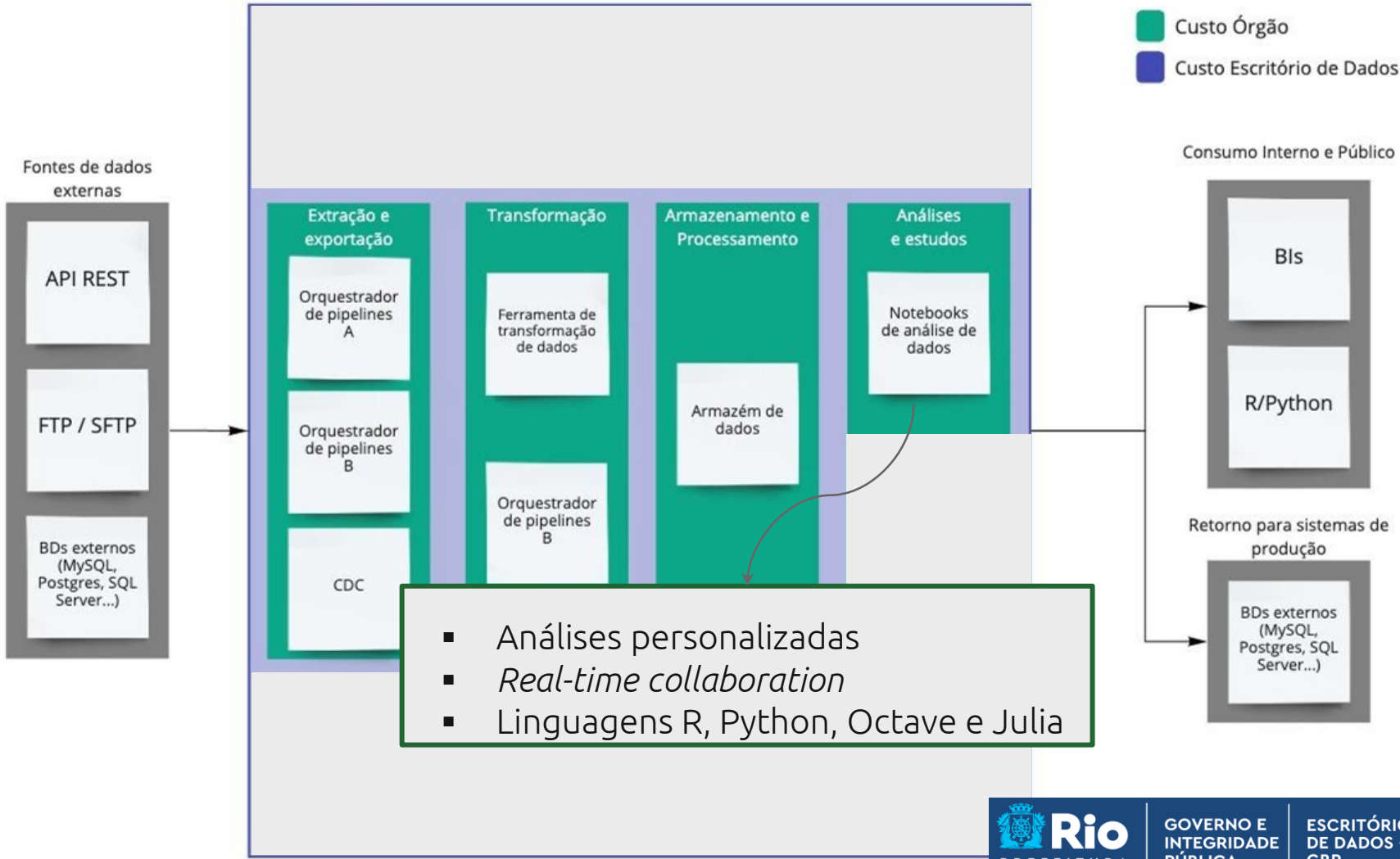
Consumo Interno e Público



- Qualquer tipo de transformação
- Materialização de views
- Linguagem SQL
- Versionamento e execução descentralizados



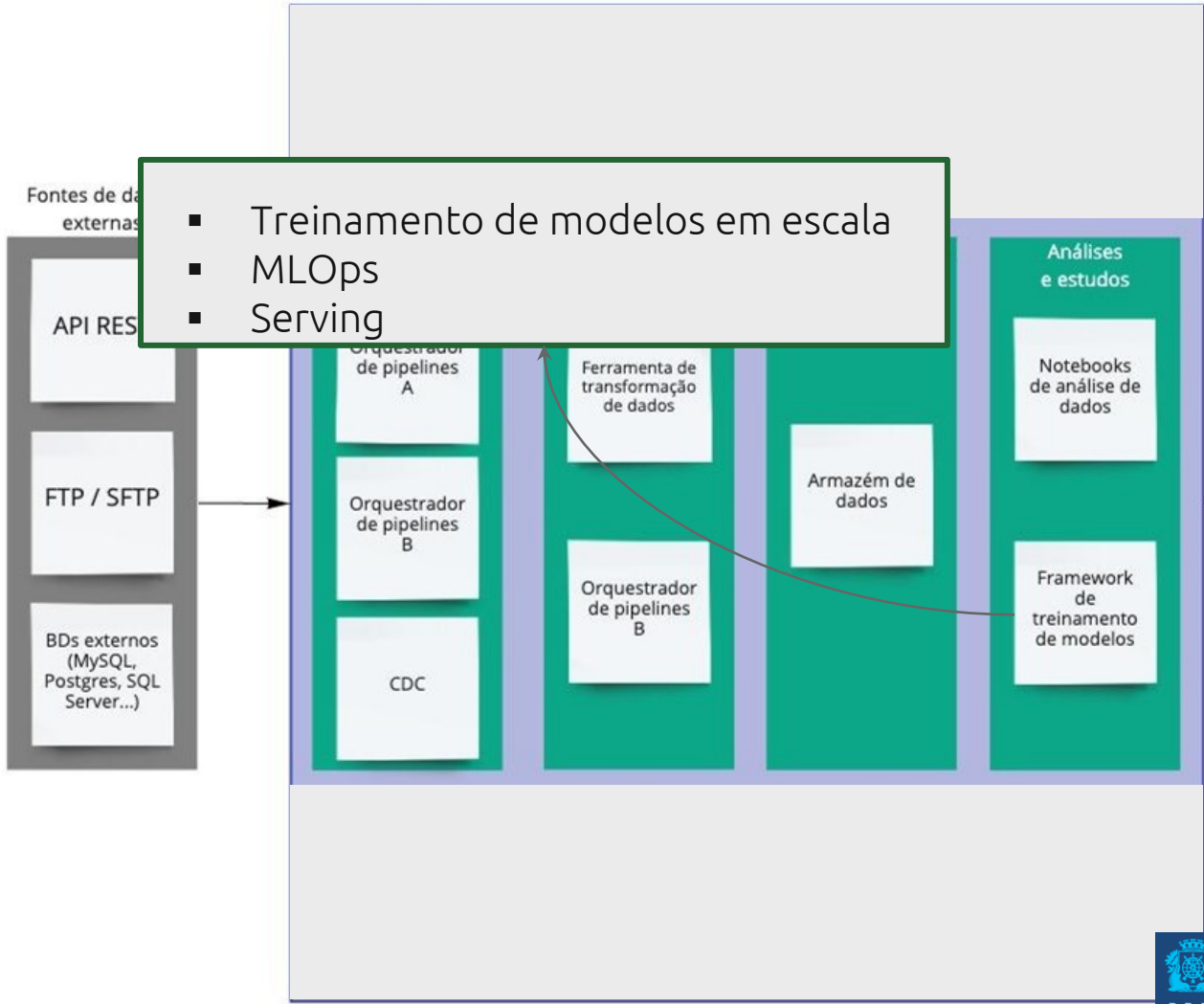


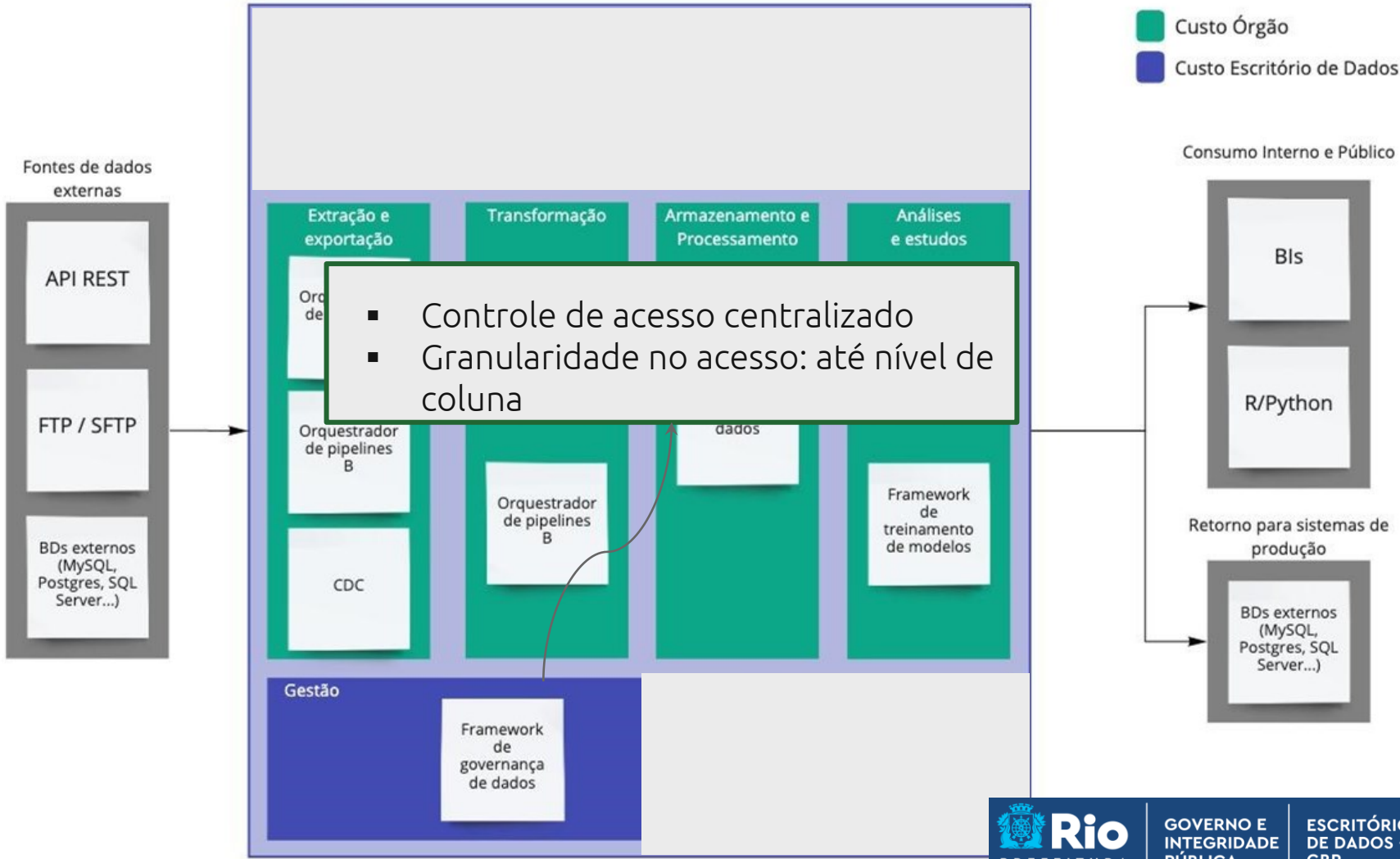


- Análises personalizadas
- *Real-time collaboration*
- Linguagens R, Python, Octave e Julia



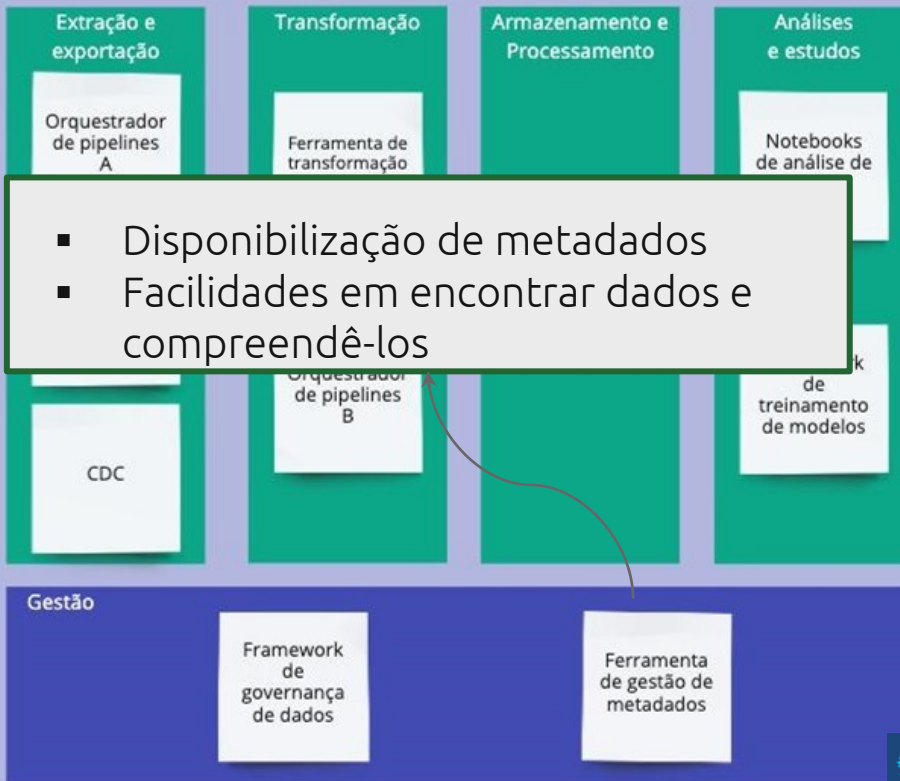
- Custo Órgão
- Custo Escritório de Dados







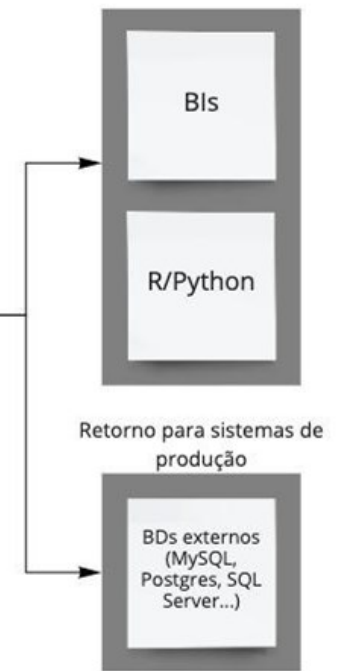
Fontes de dados externas

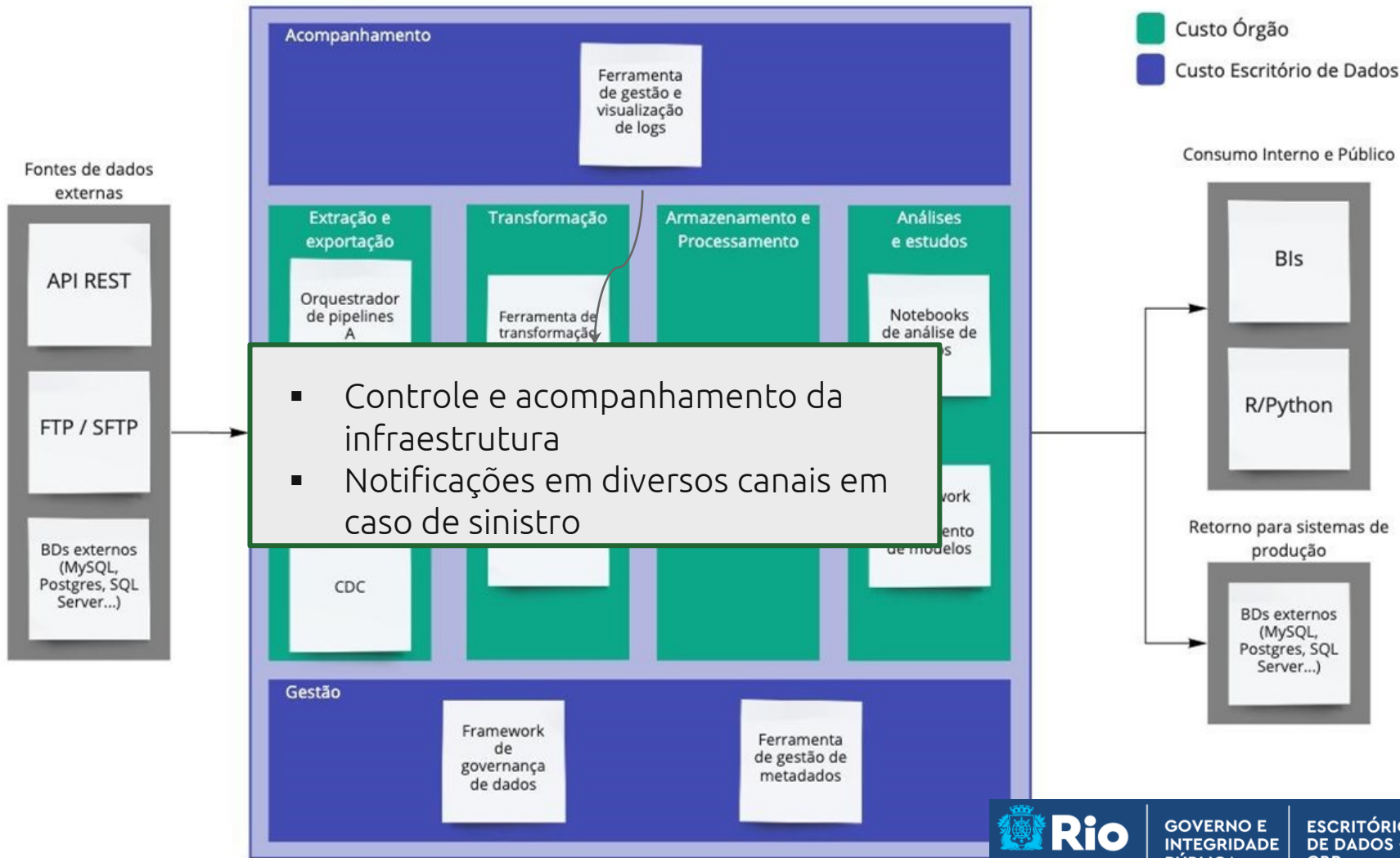


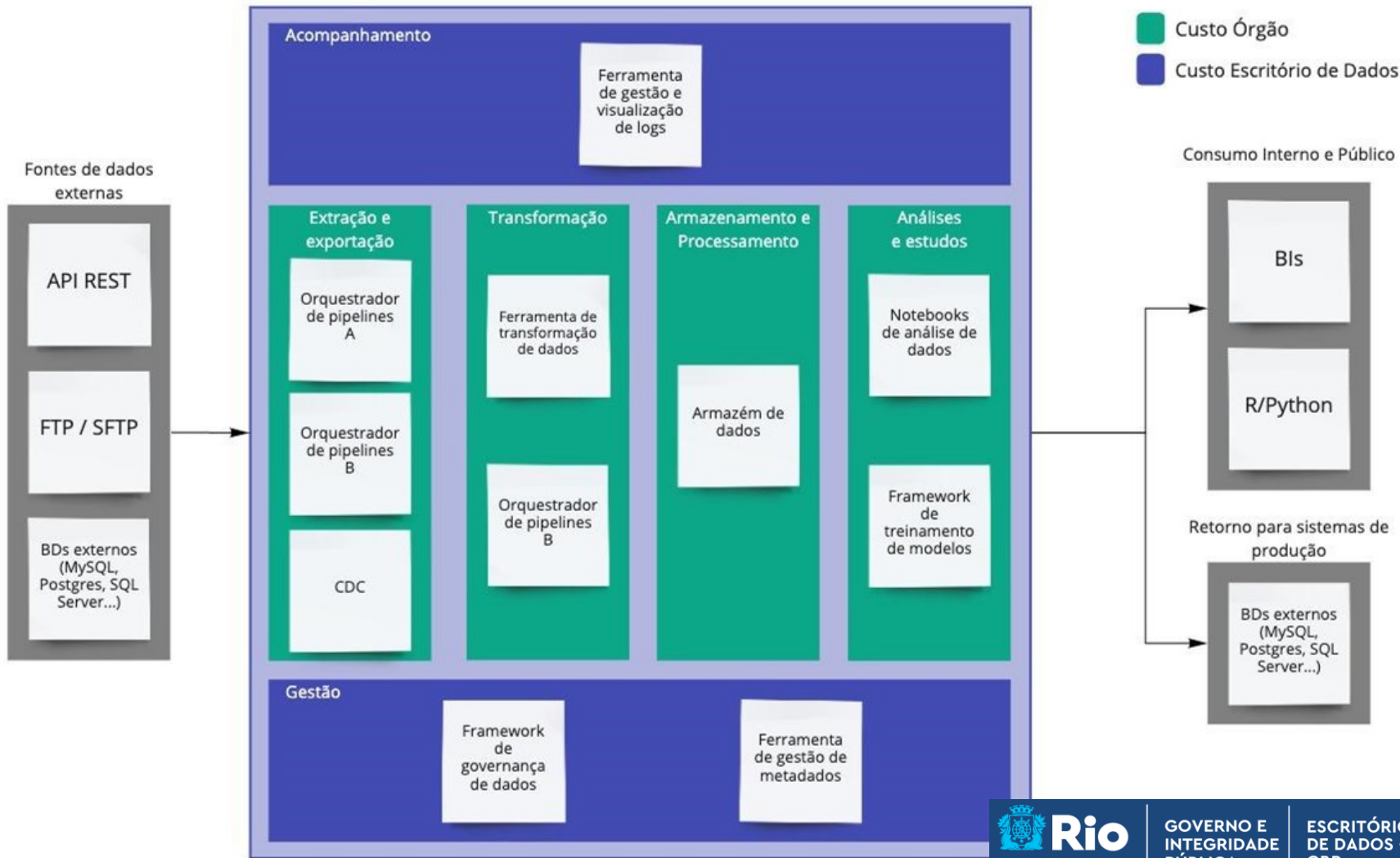
- Disponibilização de metadados
- Facilidades em encontrar dados e compreendê-los

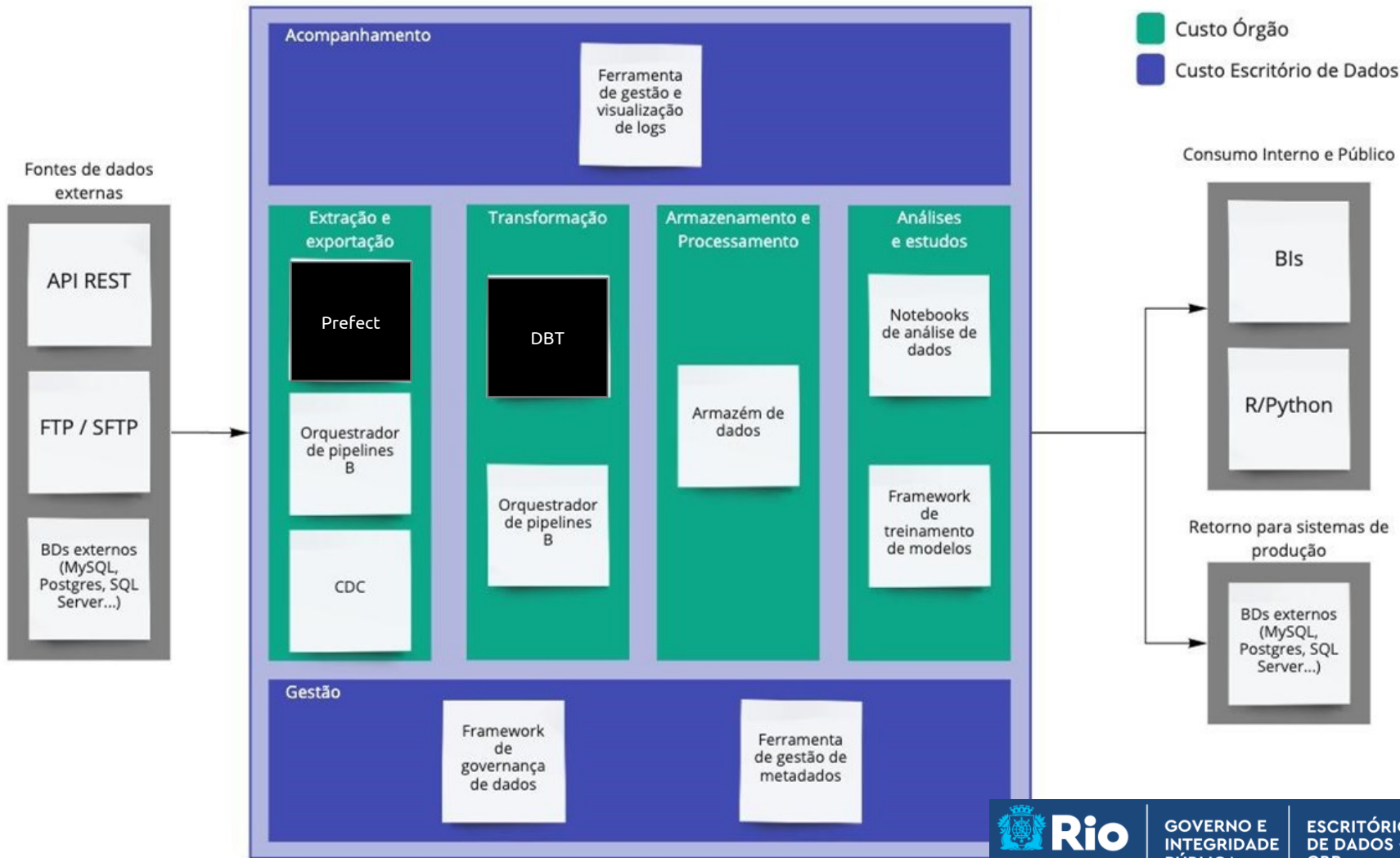
Custo Órgão  
Custo Escritório de Dados

Consumo Interno e Público



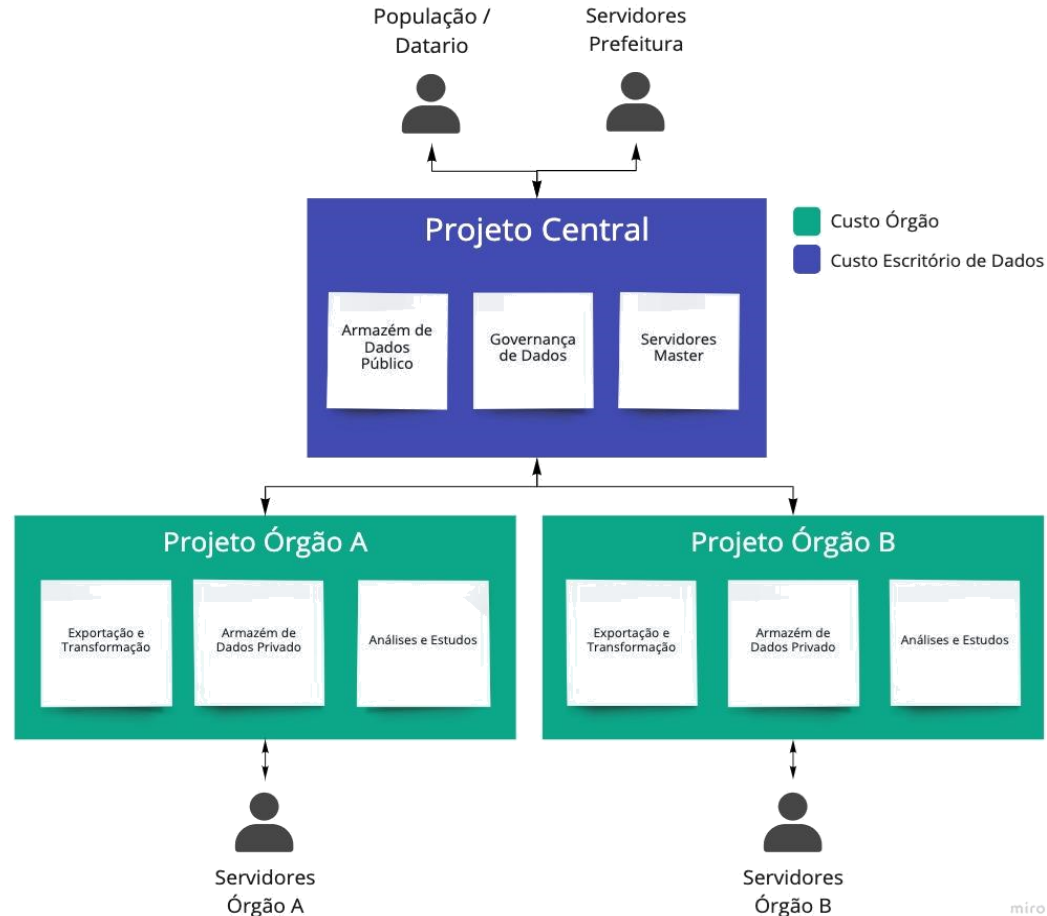






# Coordenação de componentes

- Custos de armazenamento, processamento e consulta devem ficar em uma conta ligada ao órgão
- Custos de ferramentas de gestão devem ficar ligados ao Escritório de Dados e gestores centrais
- Cada órgão deve ser capaz de gerenciar acesso aos seus dados
- Gestores centrais devem ser capazes de gerenciar todos os acessos



# Casos de uso

- Requisição de dados via API a cada minuto e processamento geográfico das informações maiores que 100GB
- Extração de dados de um banco de dados relacional com frequência diária e criação de algoritmos de ML com resultado sendo salvo em outro banco de dados relacional
- Análise de dados temporais e geoespaciais de telefonia móvel >1TB
- Obtenção e análise de dados em imagem e vídeo



# Próximos Passos

- Publicação ainda em dezembro de 2021, da Consulta Pública com esta apresentação e [catálogo de requisitos](#) para que sejam avaliados pelos fabricantes de tecnologia alinhados com o tema.
- Envio de resposta dos principais fabricantes com sua visão da arquitetura de cloud com custos calculados até 10/01/2022
  - Se possível, apresentar arquitetura com todos os componentes da empresa e arquitetura mista com componentes Open-Source sugeridos



# Anexo I – Requisitos (link [aqui](#))





**COPIE ESSA PLANILHA PARA PREENCHER**

Função	Descrição	Relevância (maior, mais relevante)	Componentes Open-Source sugeridos	Componentes Open-Source (Exemplo)	Fatores para avaliação	A solução atende ao requisito?	Estimativa de carga	Proposta de componentes 1 (Exclusivamente provedor)	Estimativa de custo 1 (Exclusivamente provedor)	Comentários 1	Proposta de componentes 2 (Provedor + Component Open-Source sugeridos)	Estimativa de custo 2 (Provedor + Component Open-Source sugeridos)	Comentários 2
Captura de dados alterados (CDC)	Rastrear alterações de dados na origem, replicando-as para o armazém de dados, mantendo os dados sincronizados.	2	-	Debezium	Não deve criar triggers adicionais na origem Deve interagir com Armazém de Dados Diversas fontes (mínimo PostgreSQL, SQL Server, MySQL) Não deve submeter consultas ao SGBD.		Backlog: 1TB Atualização: 100GB/mês						
Orquestrador de Pipelines A	Desenvolvimento de ETLs através da linguagem Python	3	Prefect	Prefect	Capacidade de uso pleno da linguagem Python a ferramenta ou framework não pode impor qualquer limitação ao código Mínimo: a transição de um Python script qualquer para uma pipeline em produção deve envolver poucas alterações de código Capacidade de agendamento de execuções Execução descentralizada (separação dos custos de computação) Capacidade de executar múltiplas pipelines a cada minuto Capacidade de obter logs detalhados das execuções Capacidade de interação com Slack, Discord, Telegram Interface web madura e intuitiva para acompanhar execução das pipelines Gestão de acesso da interface web para criação de grupos de usuários Autenticação de usuários para a interface web Gestão de pipelines por ambientes.		Início: 5 pipelines tipo A (a cada 1 min, 10 passos, ~0,5passo, ~0,5GB RAM) 10 pipelines tipo B (a cada 24 horas, 15 passos, 3passo, ~1GB RAM) 2 pipelines tipo C (a cada 24 horas, 5 passos, 15min/passo, 8GB RAM) 3 pipelines tipo D (a cada 1 hora, 10 passos, 3passo, ~0,5GB RAM) Média - 13h CPU/ida						
Orquestrador de Pipelines B	Desenvolvimento de ETLs através de uma interface gráfica	2	-	CDAP / Apache Nifi	Capacidade de desenvolvimento sem qualquer entendimento de programação Integração com diversas fontes de dados (REST APIs, bancos de dados, etc.), inclusive o próprio data lake Capacidade de agendamento de execuções Gestão de acesso aos usuários Gestão das pipelines por projeto/tema Execução descentralizada (separação dos custos de computação)		Início: 3 pipelines tipo B 2 pipelines tipo C Média - 2,5h CPU/ida						
Ferramenta de transformação de dados	Versionamento e execução de pipelines de dados em linguagem SQL	3	DBT	DBT	Capacidade de verificação de dependência entre tabelas Diferentes tipos de materialização (mínimo: tabela nativa, view ou materialização incremental) Parametrização de queries Separação de schemas		Início: 5 execuções tipo A (a cada 1 hora, duração máxima ~3min, 5GB processados) 10 execuções tipo B (a cada 24 horas, duração máxima ~50min, 25GB processados)						
Armazém de dados	Armazenamento e processamento de dados em larga escala	4	-	Ecosistema Hadoop	Capacidade de processamento e armazenamento ilimitada Otimização de processamento/custos via particionamento e compactação Interação com o banco via SQL Funções de análise nativa (SQL) <a href="https://www.w3schools.com/sql/">https://www.w3schools.com/sql/</a> Funções de análise de texto com regex Funções de análise espacial Conexões com plataformas de desenvolvimento de dashboards (PowerBI, Apache Superset) Funções de análise temporal (moving window, timezone) Suporte para sincronizar Gerenciamento automático e rápido de recursos para qualquer tipo de processamento Custos de processamento de responsabilidade do usuário Gestão de permissão de acesso aos dados para diferentes tabelas Gestão de permissão de acesso aos dados para diferentes tabelas ACID compliant Possibilidade de permitir que público acesse os dados Gestão de permissão de acesso por grupo de usuário Gestão de permissão de acesso por grupo de usuário		Início: processamento diário: 1TB armazenamento: 1TB Perspectiva (6 meses): processamento diário: 5TB armazenamento: 5TB						
Ferramenta de gestão e visualização de logs	Armazenamento e acompanhamento de logs de diversos serviços e aplicações	2	-	Elastic Stack	Capacidade de capturar logs de diversas fontes Diferentes níveis de mensagem (debug, informação, aviso, erro, etc.) Capacidade de redimensionamento automático (caso necessário) para atender a grandes rajadas de mensagens sazonais Facilidade de identificação de origem das mensagens		Início: 50 mensagens/minuto rajada de 50 mensagens a cada 1 hora rajada de 90 mensagens a cada 24 horas Perspectiva: 200 mensagens/minuto rajada de 250 mensagens a cada 1 hora rajada de 1500 mensagens a cada 24 horas						
Notebooks de análise de dados	Desenvolvimento de análises de dados interativas e replicáveis	1	-	CoCalc	Colaboração em tempo real Suporte a linguagens Python, Julia e R Capacidade de dimensionar as instâncias de execução de kernels Suporte a GPUs Suporte a formatação Markdown e LaTeX Suporte a adição de comentários no notebook (revisão) Versionamento e "blame"		Início: 16h/ida kernel tipo A (4 CPUs, 8GB RAM) 8h/ida kernel tipo B (8 CPUs, 32GB RAM) Perspectiva (6 meses): 96h/ida kernel tipo A 24h/ida kernel tipo B 4h/ida kernel tipo C (40 CPUs, 128GB RAM, 1 GPU)						
Framework de treinamento de	Pipelines de treinamento e serving	1	-	MLflow / Sekton Core	Implementação de pipelines em Python Suporte a diversos frameworks (mínimo Tensorflow, Torch e SKIL Learn) Disponibilização de modelos para inferência via HTTP-REST API		Início: - Treinamento: 24h/mês instância tipo A (8 CPUs, 32GB RAM, 1 GPU) - Serving inferência via HTTP: N/A Perspectiva (6 meses):						

Modelos	de modelos em escala			Capacidade de dimensionar as instâncias de treinamento e serving (referência)	<ul style="list-style-type: none"> <li>1 treinamento</li> <li>90h/mês instância tipo A</li> <li>72h/mês instância tipo B</li> <li>(16 CPUs, 64GB RAM, 2 GPUs)</li> <li>Serving (referência via HTTP)</li> <li>24h/mês de 3 modelos de ordem de grandeza da EfficientNetB6 (~45M parâmetros)</li> </ul>
				<ul style="list-style-type: none"> <li>Suporte a redimensionamento automático de instâncias de serving</li> <li>Suporte a GPUs</li> </ul>	
Framework de governança de dados	Ferramenta capaz de gerir acesso aos dados e acompanhar linhagem dos dados	3	-	Apache Atlas	<ul style="list-style-type: none"> <li>Integração com o armazém de dados</li> <li>Interface web</li> <li>Gestão de permissão dos acessos por usuário</li> <li>Gestão de permissão dos acessos por grupo de usuário</li> <li>Granularidade de acesso a nível de coluna em tabelas do armazém de dados</li> <li>Capacidade de disponibilização pública dos dados</li> <li>Versionamento e "Name"</li> </ul>
					<ul style="list-style-type: none"> <li>Início:</li> <li>8 administradores</li> <li>32 usuários</li> <li>Acesso público 10 pessoas/mês</li> <li>Perspectiva (6 meses):</li> <li>24 administradores</li> <li>72 usuários</li> <li>Acesso público 500 pessoas/mês</li> </ul>
Ferramenta de gestão de metadados	Ferramenta para gestão e busca de metadados	1	-	Apache Atlas	<ul style="list-style-type: none"> <li>Interação com o armazém de dados</li> <li>Interface web</li> <li>Geração automática de documentação</li> <li>Funcionalidades de busca de dados através de metadados</li> <li>Versionamento e "Name"</li> </ul>
					<ul style="list-style-type: none"> <li>Início:</li> <li>Perspectiva (6 meses):</li> </ul>
Geral	Requisitos gerais da arquitetura	4	-		<ul style="list-style-type: none"> <li>Número total de componentes da solução</li> <li>Solução em Cloud</li> <li>Disponibilidade de servidores no Brasil</li> <li>Número de datacenters no Brasil</li> <li>Certificado ISO 27001 e 27002 para todos os componentes</li> </ul>
Observações Gerais					
+ Custo não relacionado a componentes					
<b>Custo Total Estimado:</b>					