

SystemC e Modelos em Nível Transacional

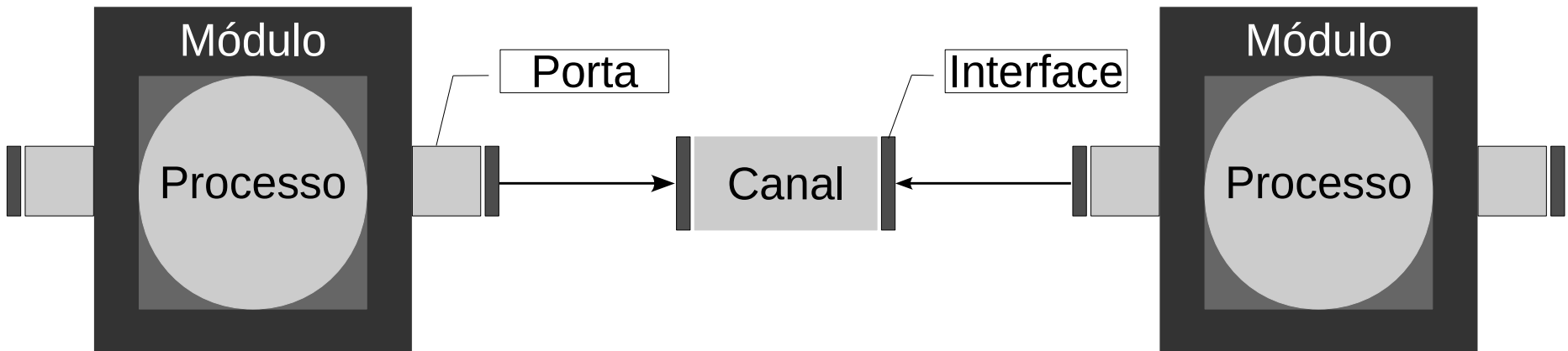
Tiago Trindade da Silva
tiago@linux.com

Programa

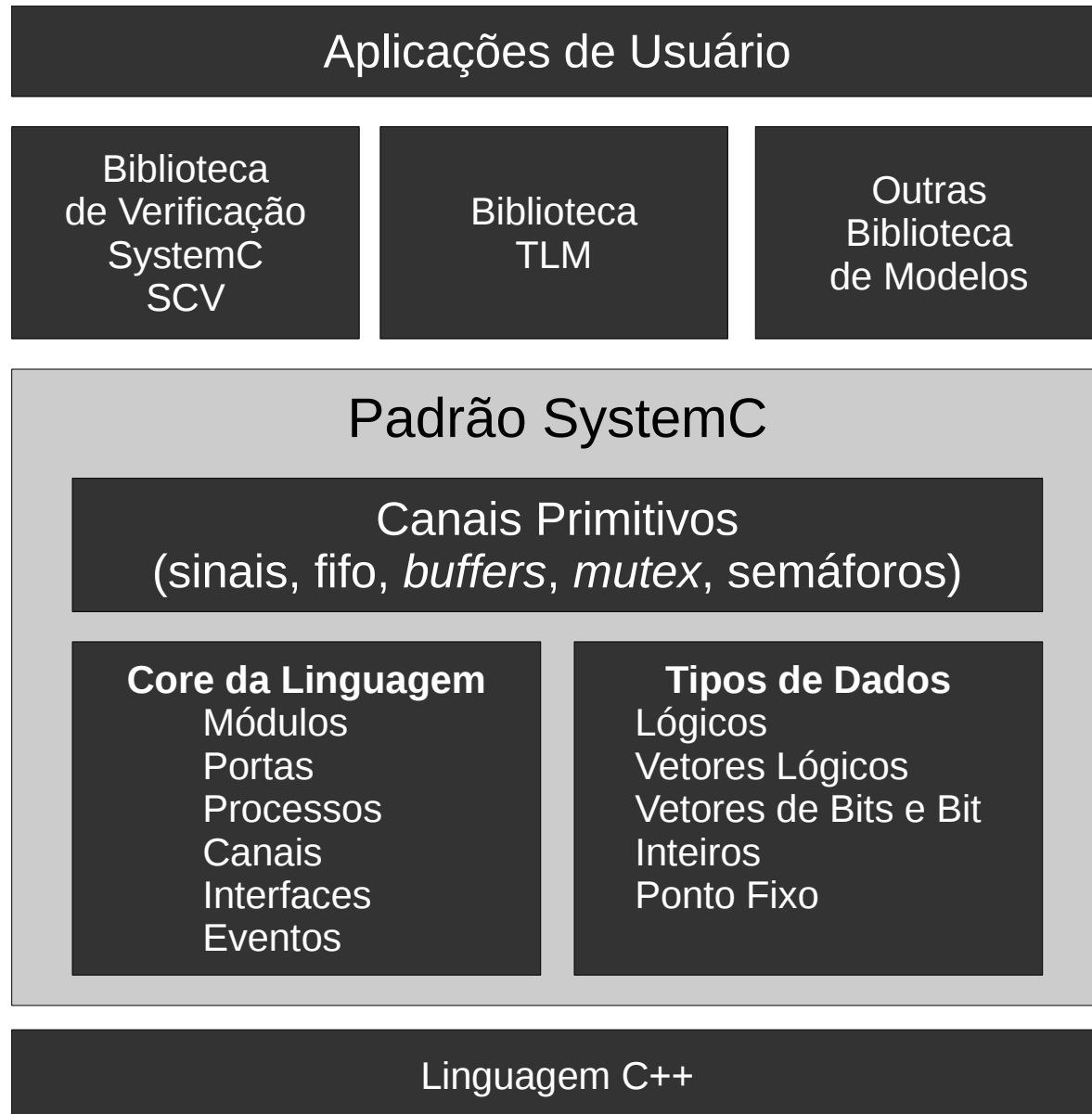
- Objetivos:
 - Apresentar uma introdução ao *Desing* em Nível Transacional em SystemC.
- Tópicos Abordados:
 - Introdução e *Background*
 - O mundo do SystemC
 - TLM
 - Estilos de Códigos

Recursos do SystemC

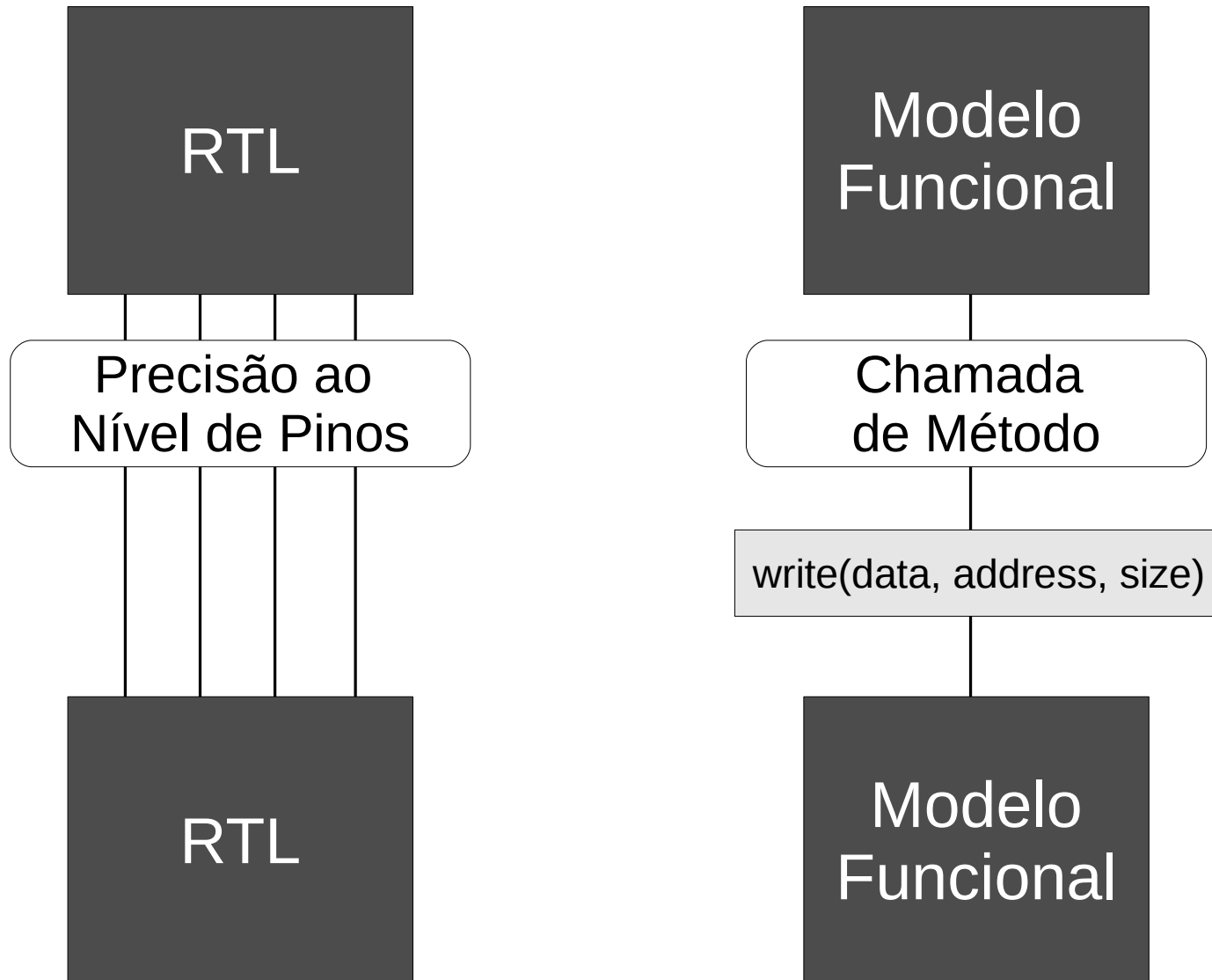
- Módulos (estrutura)
- Portas (estrutura)
- Processos (computação, concorrência)
- Canais (comunicação)
- Interfaces (refinamento da comunicação)
- Eventos (temporização, escalonamento, sincronização)
- Tipos de dados (*Hardware*, ponto fixo)



Arquitetura do SystemC



Modelo Transaccional



Características das Plataformas Virtuais

- Precisão de registradores
- Funcionalidade completa
- Sem *clock*
- Sem pinos
- Implementação sem detalhes
- Temporização aproximada ou inexistente
- Suficientemente rápida para iniciar um OS em segundos
- Disponível meses antes do modelo RTL

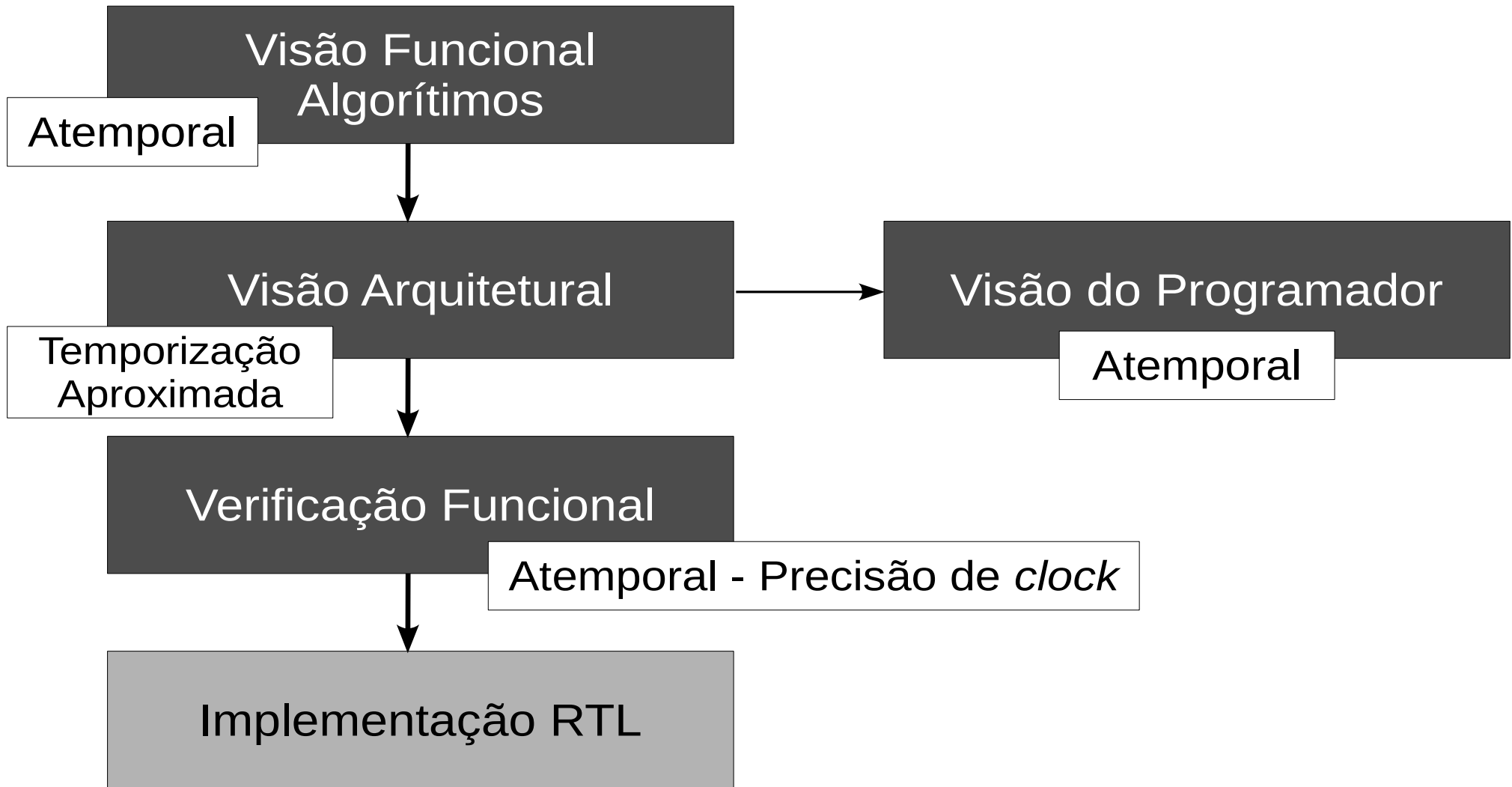
Características das Plataformas Virtuais

Simulação do Conjunto de Instruções ou <i>Software Stubs</i>	Modelo Transacional	RTL
Disponível Rapidamente	Disponível Rapidamente	Muito Depois
Rápido o suficiente para rodar aplicações	Rápido o suficiente para rodar aplicações	Muito lenta para rodar aplicações
Poucos ou nenhum detalhe de <i>hardware</i>	Precisão de Registradores	Precisão de registradores e pinos.
Sem informações de tempo	Algumas informações de tempo	Temporização no nível do ciclo de <i>clock</i>

Requerimentos TLM-2

- Modelos transacionais e barramentos mapeados em memória.
- Precisão ao nível de registradores
- Funcionalidade completa
- Temporização aproximada
- Atemporal
- Capaz de iniciar um OS rapidamente
- Troca de modelos transacionais.

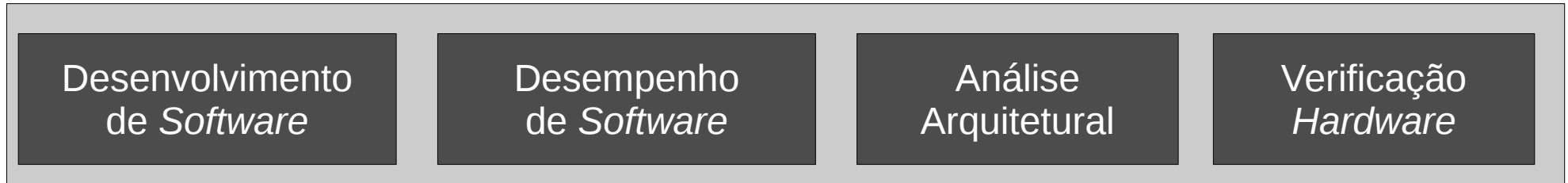
Níveis de Abstração



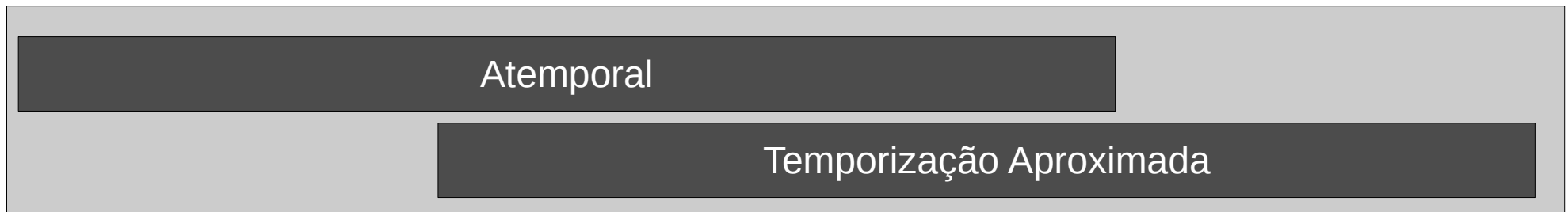
Casos de Uso

Mecanismos e Estilos de Código

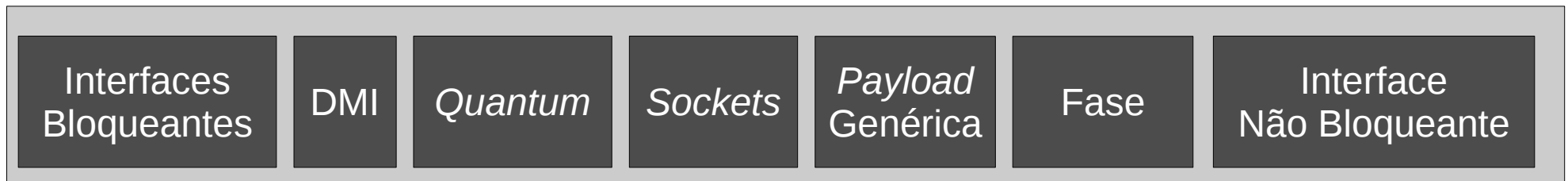
Casos de Uso



Estilos de Codificação TLM-2



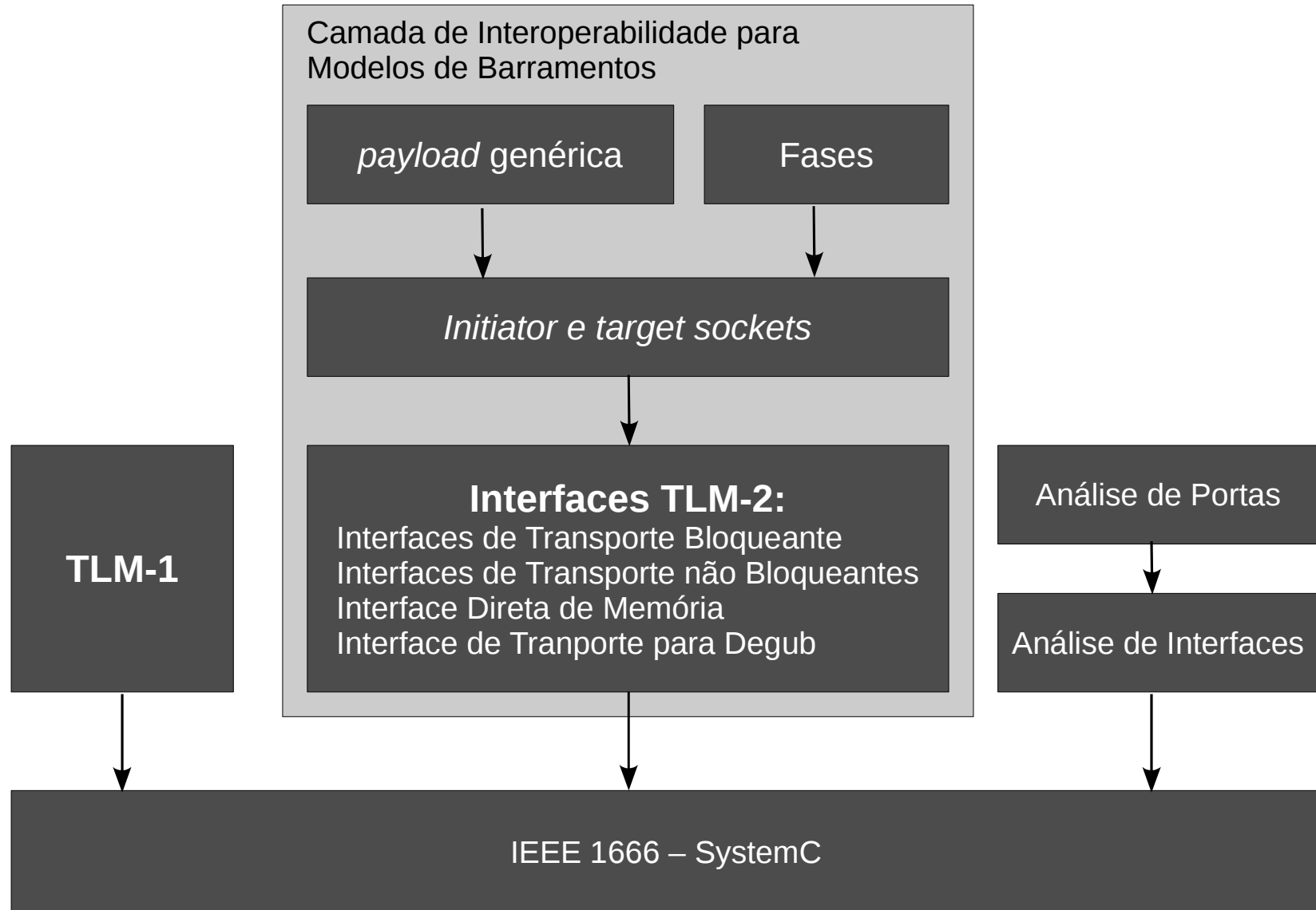
Mecanismos (API do TLM-2)



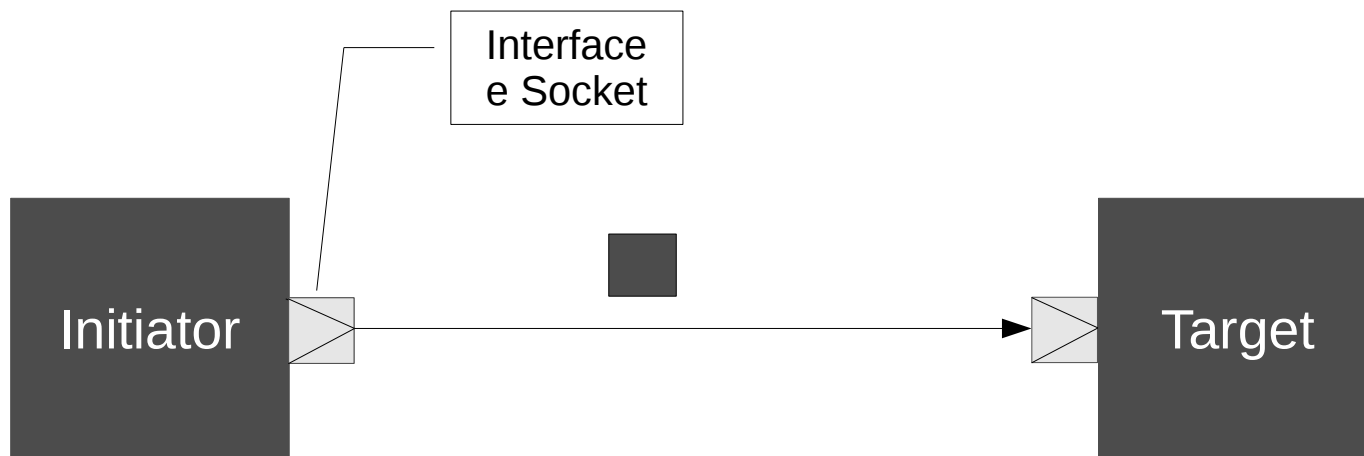
Estilos de Codificação

- Atemporal = rapidez
 - Detalhes de temporização suficientes para iniciar um OS e rodar sistemas multi-core
 - Processos podem rodar antes do tempo de temporização (desacoplamento temporal)
 - Cada transação é completada em uma chamada de método
 - Interface de Acesso Direto a Memória
- Temporização Aproximada
 - Aproximação *aka* ciclo ou precisão contagem de ciclos
 - Suficiente para exploração arquitetural
 - Processos rodam sincronizados com o tempo de simulação
 - Cada transação tem 4 pontos de temporização (extensível)

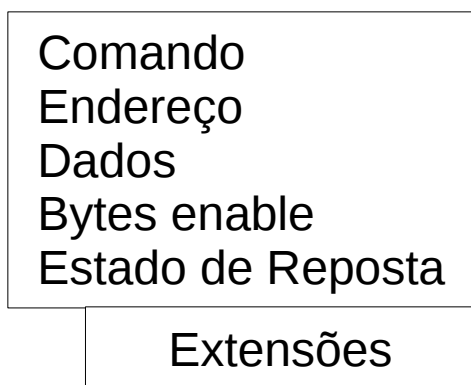
Classe TLM 2.0



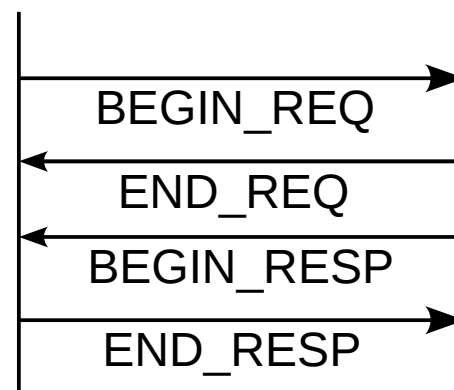
Camada de Interoperabilidade



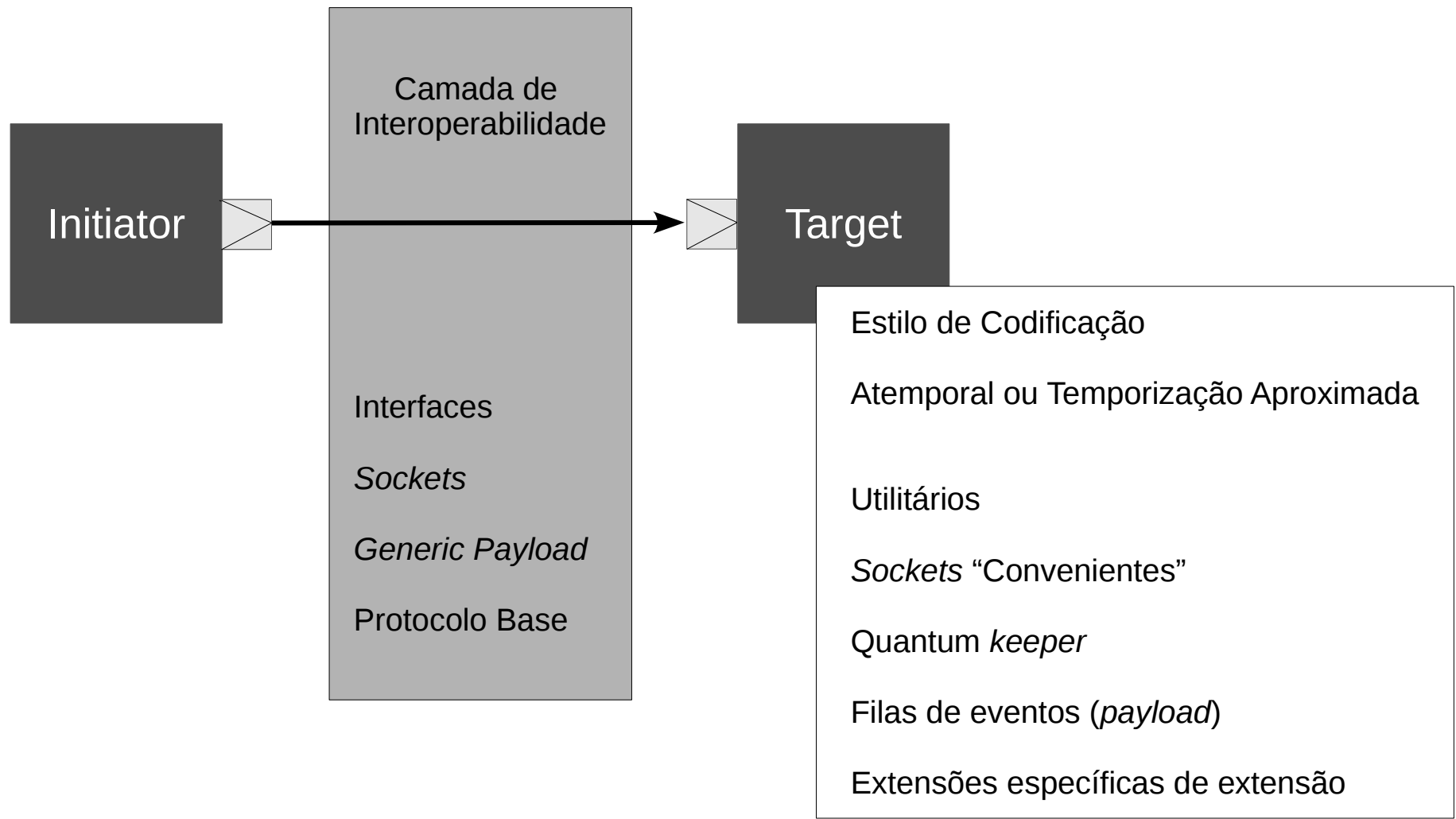
Generic Payload



Generic Payload



Utilitários



Exemplo