

ZBus do Laboratório ao Zephyr RTOS



Rodrigo Peixoto

Professor na UFAL
Centro de Inovação Edge

Patrocinado por



WEBINAR
EMBARCADOS



Patrocinado por



MOUSER
ELECTRONICS

Agenda

1. Quem é Rodrigo?
2. Ambiente de inovação
3. A origem do zbus
4. A jornada até o Zephyr RTOS
5. Onde estamos e para onde vamos?

Quem é Rodrigo? // *Linha do tempo*

2003 a 2008: Engenheiro de computação formado no Centro de Informática da UFPE

- BrazilIP — *Iniciação Científica*
- Itaotec — *Estágio*
- C.E.S.A.R. — *Estágio*
- Ministério de Educação do Brasil — *Projeto Lampejo*
- Força Aérea Americana — *Projeto 3D Router*

2009 a 2010.2: Mestre pela UFAL — *Laboratório COMPE*

2011: Concursado como professor na UFAL

Quem é Rodrigo? // *Linha do tempo*

2012 a 2013.2: Consultor de desenvolvimento de aplicativos (América Latina)

2013 a 2015: Sócio do Piggo — *Startup de troco eletrônico (banco digital em 2013)*

2015 a 2016: Apiário Terra do Nunca — *Apicultura e meliponicultura*

2016 a 2026: Centro de Inovação Edge na UFAL

10 anos de inovação // *universidade-indústria*

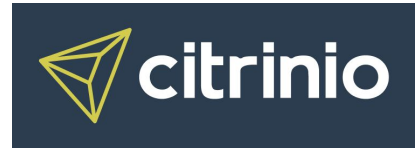
Criação da divisão de sistemas embarcados do Edge

- Laboratório de prototipação — *um oásis no deserto*
- Hoje somos +30 pessoas — *formados “dentro de casa”*
- +50 projetos executados com a indústria



Desenvolvimento do VSX — *mais uma startup*

- Sistema de monitoramento inteligente de iluminação pública
- Software complexo
 - Rede mesh Bluetooth + Celular 2G
 - Atualização de firmware
 - Requisitos de segurança



A origem do zbus

2017 a 2018: Na segunda versão do VSX os requisitos apertaram

- O bare-metal da primeira versão não funcionaria
- A modularização era chave para a grande refatoração

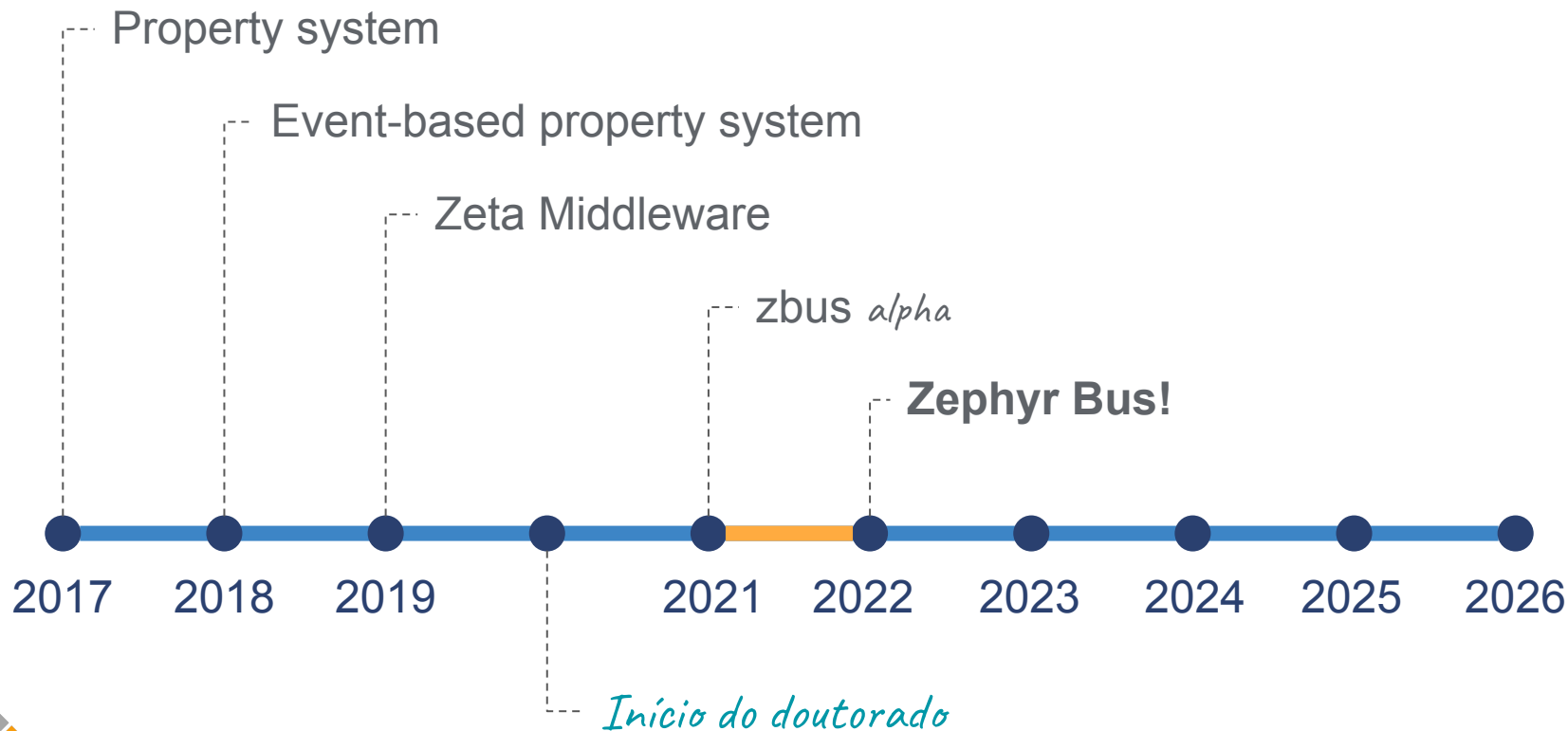
Outros projetos do laboratório já demandariam mais

- Início da era IoT
- RTOSes eram inevitáveis para nós

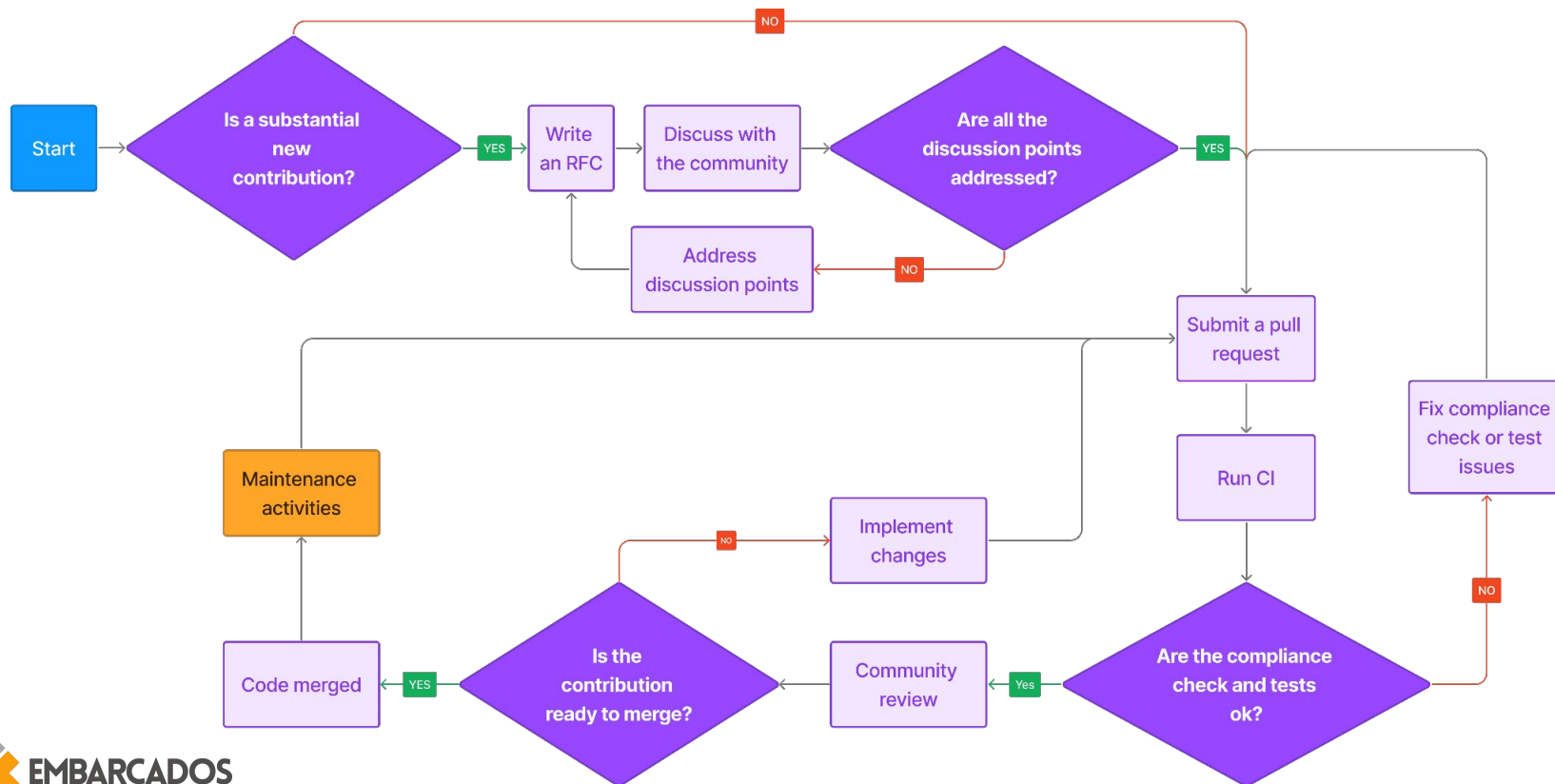
Mesmo utilizando RTOS a modularização era difícil com o básico

- Semáforos, mutexes, message queues, etc. não pareciam ajudar tanto
- C ainda é a linguagem utilizada — *desilusões com C++*

A jornada até o Zephyr RTOS



O processo de submissão



A linha do tempo da submissão



Architecture Working Group // reunião 01

Primeira sabatina

- ~1 hora de duração
- Várias empresas na reunião (inclusive o autor do EM)
 - O líder do grupo na época era o representante de PD&I da Nordic
- Muitas perguntas
- Muitas críticas
- Poucas sugestões

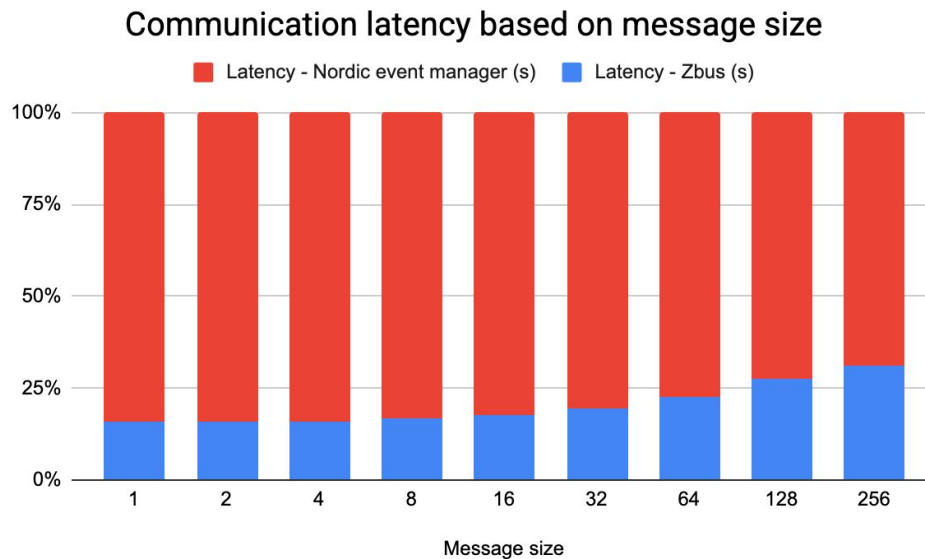
Momento de negação da comunidade...

Muito trabalho na RFC *// alterações e adições*

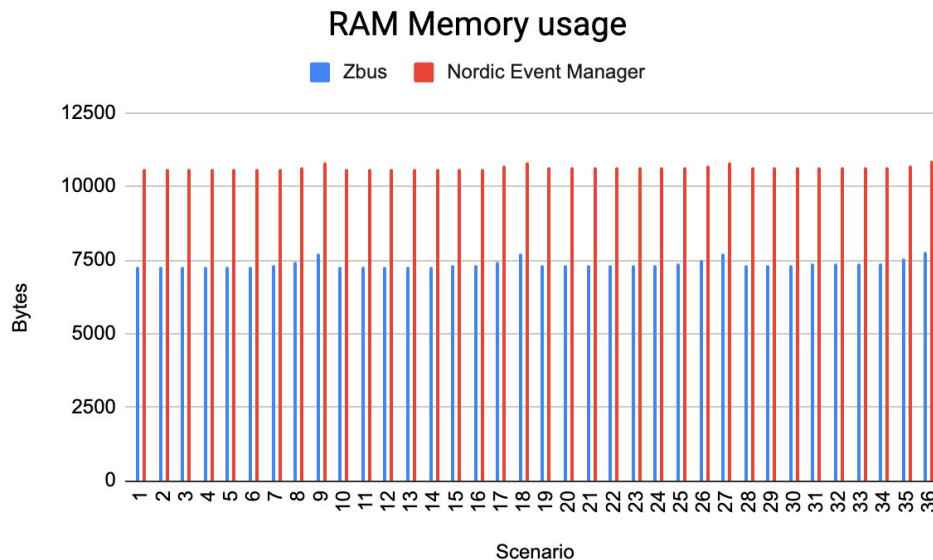
- Exploração de alternativas
- Descrição detalhada de funcionalidades
- Mapeamento de riscos
- Tabela comparativa
- Pedido de comparação entre o Event Manager e o zbus

Já estava demonstrando que eu não era um aventureiro.

Comparação com o Event Manager (EM)



Comparação com o Event Manager (EM)



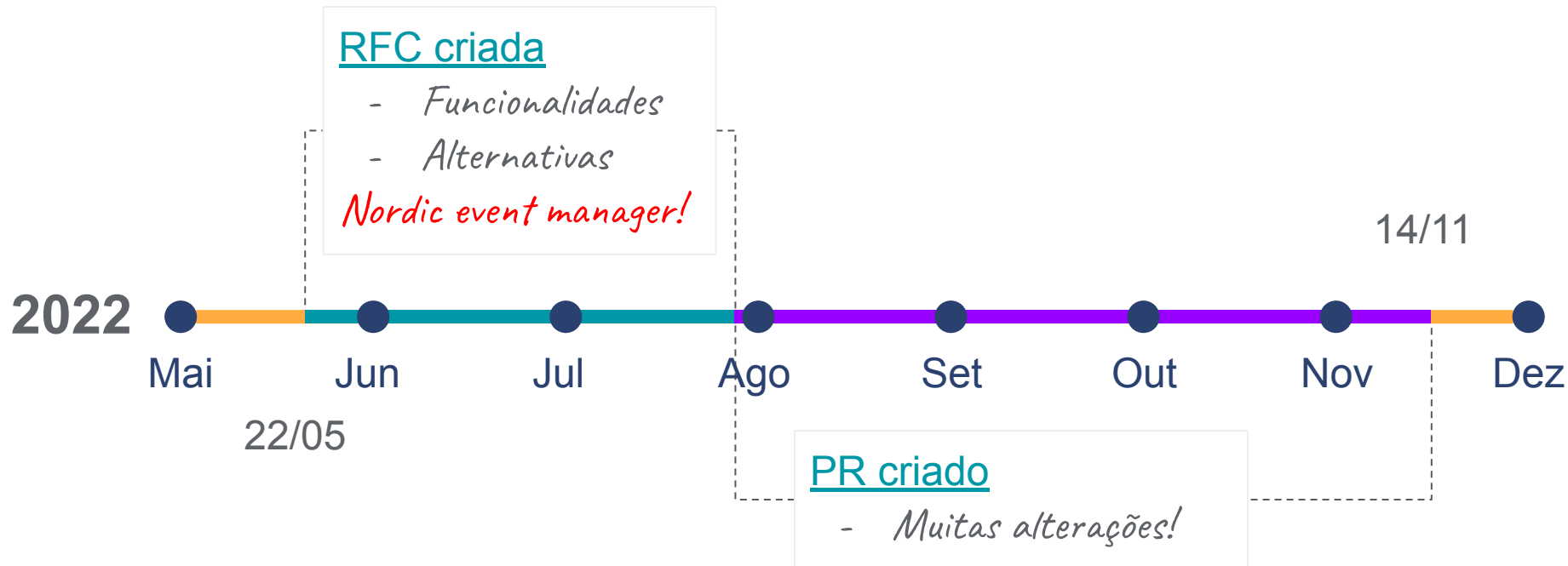
Architecture Working Group // reunião 02

Apresentação de resultado da comparação + segunda sabatina

- ~40 minutos de duração
- Várias empresas na reunião (inclusive o autor do EM)
 - O líder do grupo na época era o representante de PD&I da Nordic
- Muitas perguntas
- Algumas críticas
- Boas sugestões

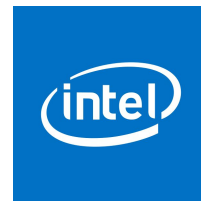
**Momento em que a comunidade aceitou a
ideia de um possível novo barramento**

A linha do tempo da submissão



Sumário da contribuição

- 5 meses e meio de duração
- 13 commits
- 432 mensagens trocadas
 - Muitos ajustes requisitados
- +5500 linhas de código
 - Implementação
 - Testes
 - Samples
 - Documentação



Resultados das discussões (RFC+PR)

- **Sair de um event dispatcher centralizado para um descentralizado**
 - Criação do VDED
- Adição dos listeners
- Utilização de mutex ao invés de semáforos
- Definição e subscrição de canais descentralizadas
- Observações em tempo de execução
- ~~Desbloqueio dos canais antes da conclusão das notificações~~ *negado*

ZBus observers



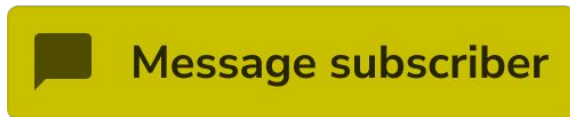
Callback – *sync, rápido e leve (executado no contexto da publicação)*



Offloaded callback – *async, rápido (executado fora do contexto da publicação)*

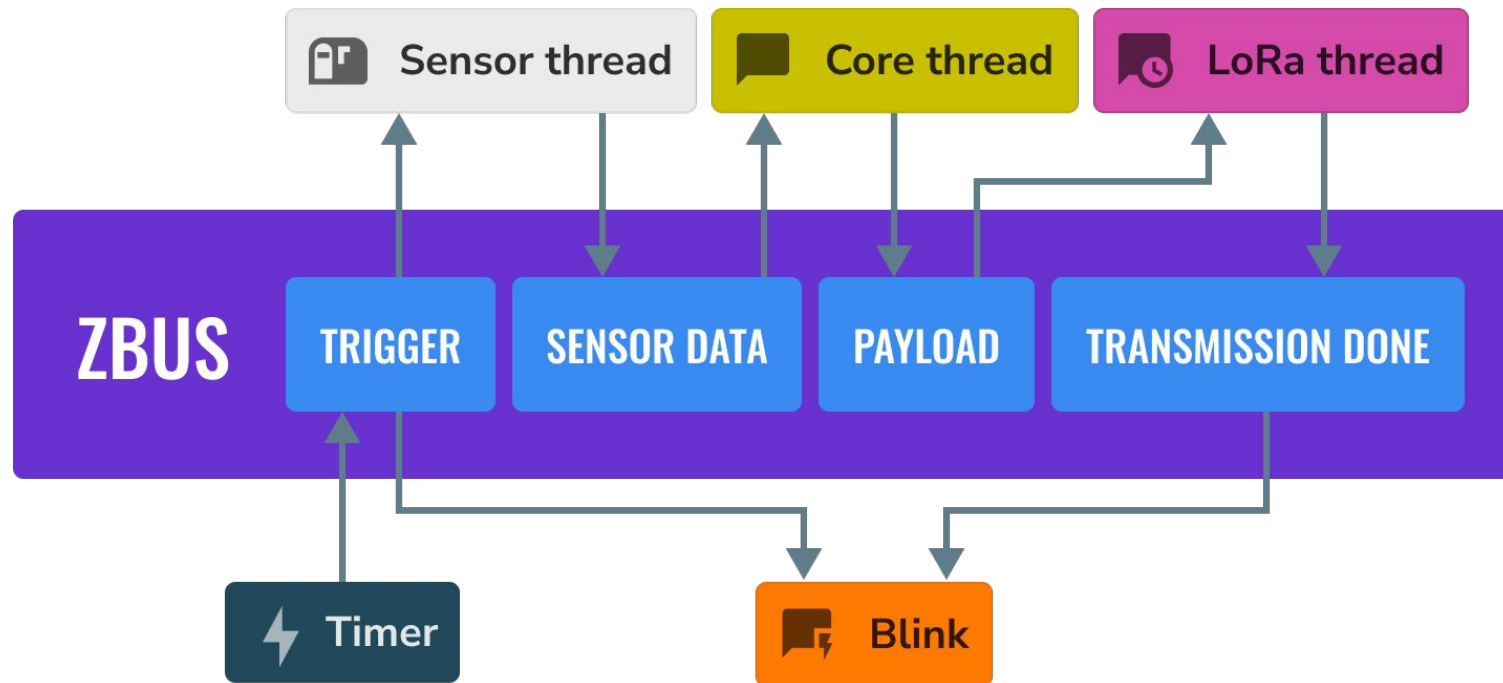


Notificação de mudança – *async, leve e sem garantia de entrega*

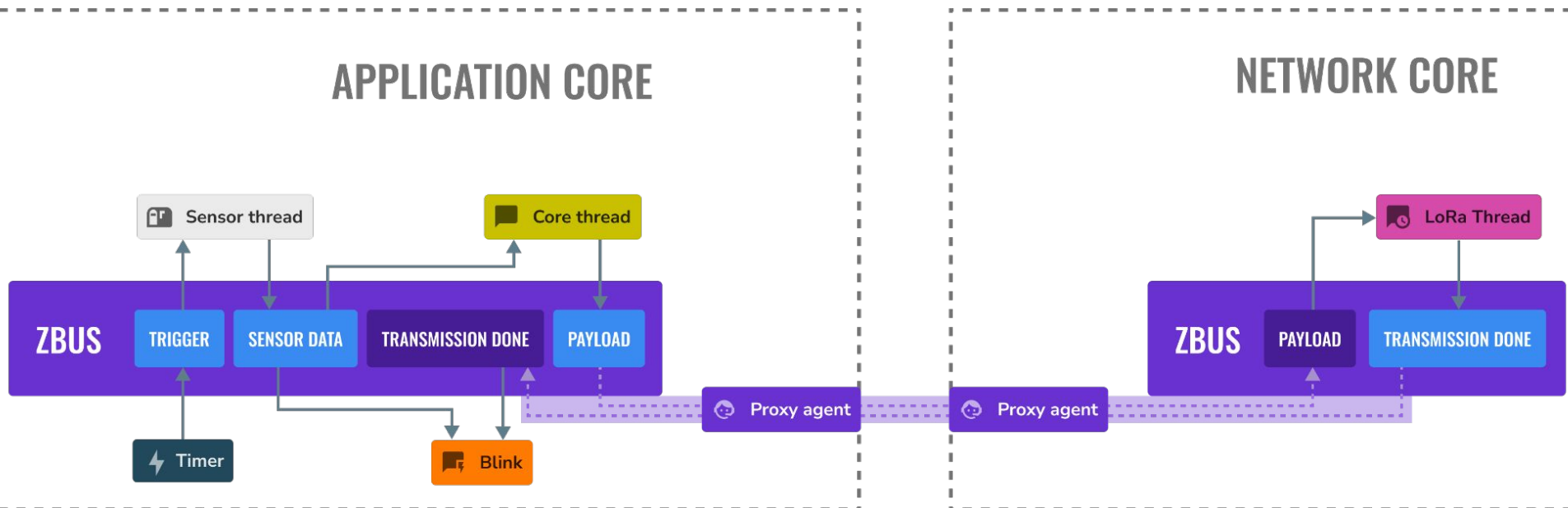


Mensagem – *async e com garantia de entrega*

ZBus exemplo

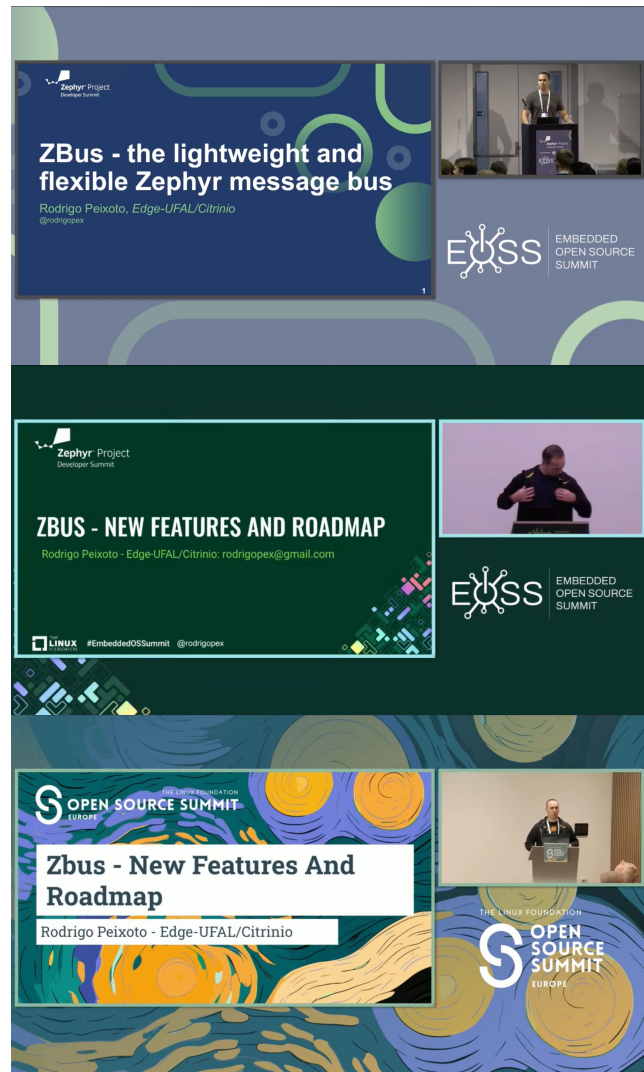


ZBus multi-core (AMP)



O que aconteceu após o merge?

- Mantenedor do zbus
- Comecei a ser conhecido na comunidade
- Canal no Discord do Zephyr dedicado ao zbus
- Palestras em eventos do Zephyr (Praga em 2023, Seattle em 2024, Amsterdã em 2025)
- Coordenação do desenvolvimento de novas features junto a grandes empresas (Nordic e Multi-core zbus)



O que aconteceu após o merge?

- Treinamentos de Zephyr para empresas dos EUA e Europa
- Eleito pela comunidade membro do TSC do Zephyr
- Consultorias intensificadas, principalmente, na Europa
- Internacionalização das atividades do Edge



aC6

Lições aprendidas no processo

- Paciência e perseverança sempre
- Somos melhores do que imaginamos – *síndrome do cachorro vira-lata*
- Conhecimento sólido no tema é imprescindível – *o doutorado me forçou a isso*
- Saber ouvir, filtrar e entender feedback e sugestões é difícil – *não se apaixone pelo código ou ideia*
- Muita gente/empresa contribui pensando em si e não no bem maior



Próximos passos

- Melhorias de desempenho no zbus
- UART *backend* para o *proxy agent*

Outros projetos pessoais:

- **Zephlets** — *Lightweight service subsystem for Zephyr RTOS. A modular architecture enabling simple, event-driven services built on zbus for inter-service communication.*
- **Objective-Z** — *a simplified, ahead-of-time compiled subset of Objective-C for Zephyr RTOS that adds static-dispatch, heap-free object-oriented features on top of plain C.*
- **Zephyr Virtual Board** — *under development*

<https://github.com/rodrigopex/>



Contatos



Rodrigo Peixoto



rodrigopex

rodrigopex@gmail.com

OBRIGADO!



**MOUSER
ELECTRONICS**

Patrocinado por



www.embarcados.com.br



linkedin.com/embarcados



[@portalembarcados](https://instagram.com/portalembarcados)



[youtube/Embarcados TV](https://youtube.com/EmbarcadosTV)