

Qt



Qt Inside Telefonica Vivo

Milton Soares Filho
milton.sfilho@telefonica.com

Samuel Mello
samuel.mello@telefonica.com





10615

Chamado Técnico para Vivo TV



Milton Soares Filho

Cientista da computação formado pela UFPR, trabalhando com C e Linux desde 1998.

Ex-Conectiva e participante nos primórdios da evolução do framework Qt4 durante período no INdT (OpenBossa). Passando por desenvolvimento para mobiles e set-top-boxes, atualmente trabalha na integração dos middlewares para IPTV do grupo Telefonica (Brasil, Chile, Argentina, Colombia, etc).



@miltonsoaresfilho



Samuel Mello

Cientista da computação com mestrado em redes de computadores pela UFPR.

Tem dedicado os últimos 14 anos ao desenvolvimento de software para indústria de telecom, principalmente em dispositivos embarcados e atualmente é arquiteto do software embarcado em Set-Top-Boxes IPTV atualmente usado por mais de 100 mil usuários no grupo Telefonica, que utiliza bibliotecas Qt tanto para interface de usuário quanto para interação com plataformas e outros módulos de middleware.

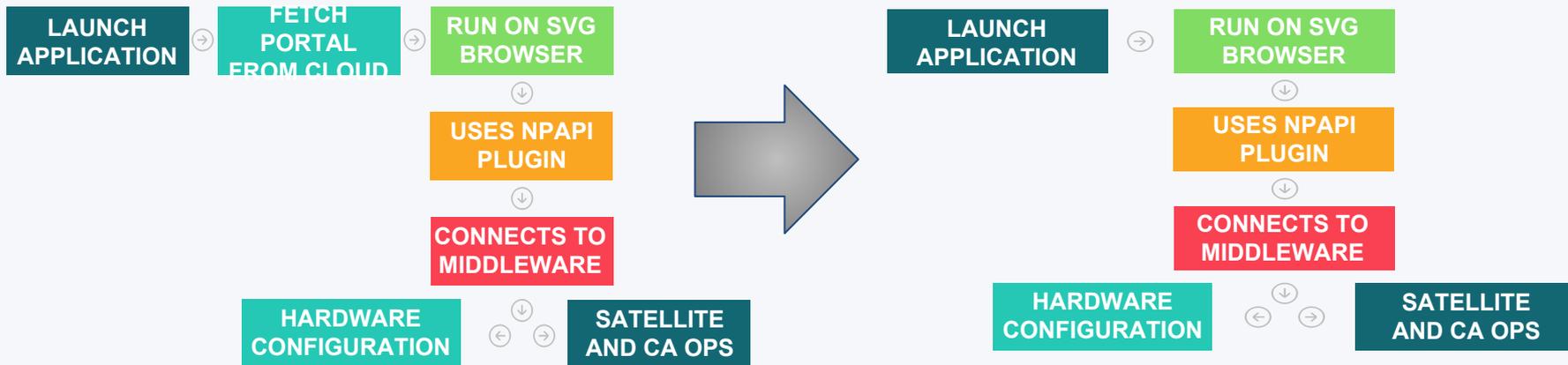


@svmello

Criação Equipe Desenvolvimento GVT



- ◆ Equipe pequena iniciada na GVT a partir de 2012
- ◆ Inicialmente fazia somente integração de ferramentas de terceiros
- ◆ UI feita em browser SVG + javascript
- ◆ Gradualmente substitui layout na nuvem por solução stand-alone
- ◆ Evolução do projeto usava HTML-SVG + Angular
- ◆ Redução drástica da quantidade de chamados técnicos realizados

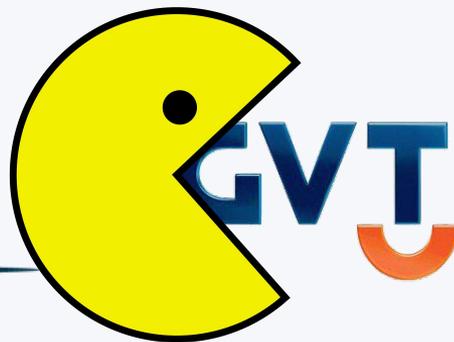


Aquisição pela Telefonica



- ◆ Telefonica, empresa hispanica-mundial fundada em 1924
- ◆ Mais de 10000 funcionários pelo mundo
- ◆ Políticas agressivas de down-sizing
- ◆ Atuação em Telecom Fixa focada inicialmente no estado de São Paulo
- ◆ Nossa equipe praticamente somente com especialistas
- ◆ Concentração de operações em São Paulo ao invés de Curitiba
- ◆ Desenvolvimento tecnológico em Madri para toda América Latina

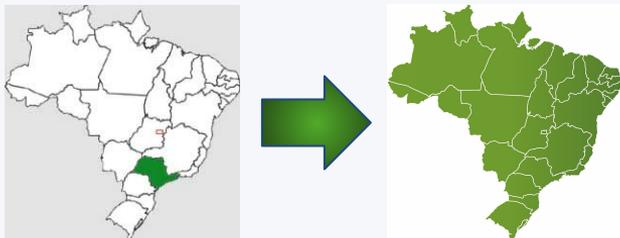
Telefonica



Projeto OpenPlatform



- ◆ Aquisição da GVT possibilita expandir o serviço de IPTV para todo o Brasil, porém plataforma existente não seria adequada.



Comprar solução pronta de um integrador ou integrar in-house?

- ◆ Telefonica atuaria como integradora da solução
- ◆ Arquitetura aberta: escolha componente a componente
- ◆ Redução de custos
- ◆ Melhor time-to-market
- ◆ Maior controle da solução



- ◆ Nosso time ficou responsável pelo desenvolvimento do software embarcado a ser usado no STB

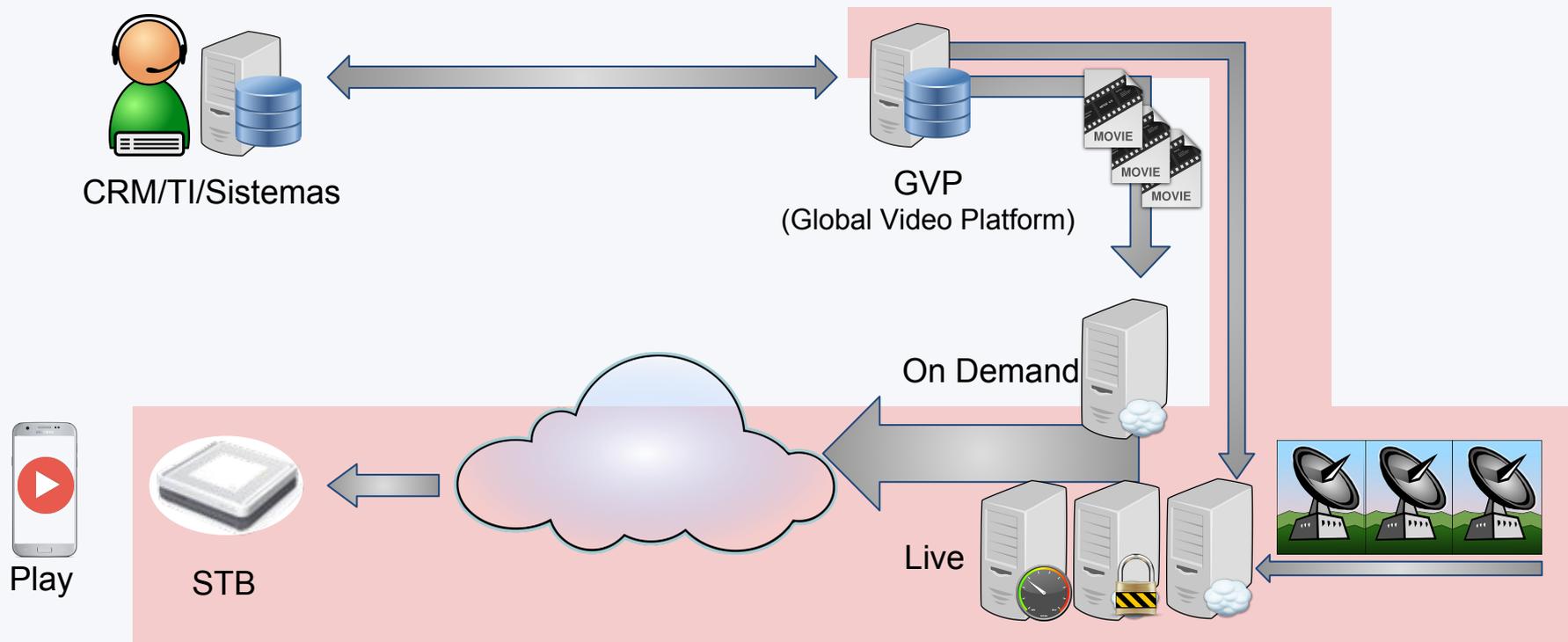


ENLISTED TO A PROJECT THATS GONNA
ENGULF SOCIAL LIFE FOR 2 YEARS
STRAIGHT..

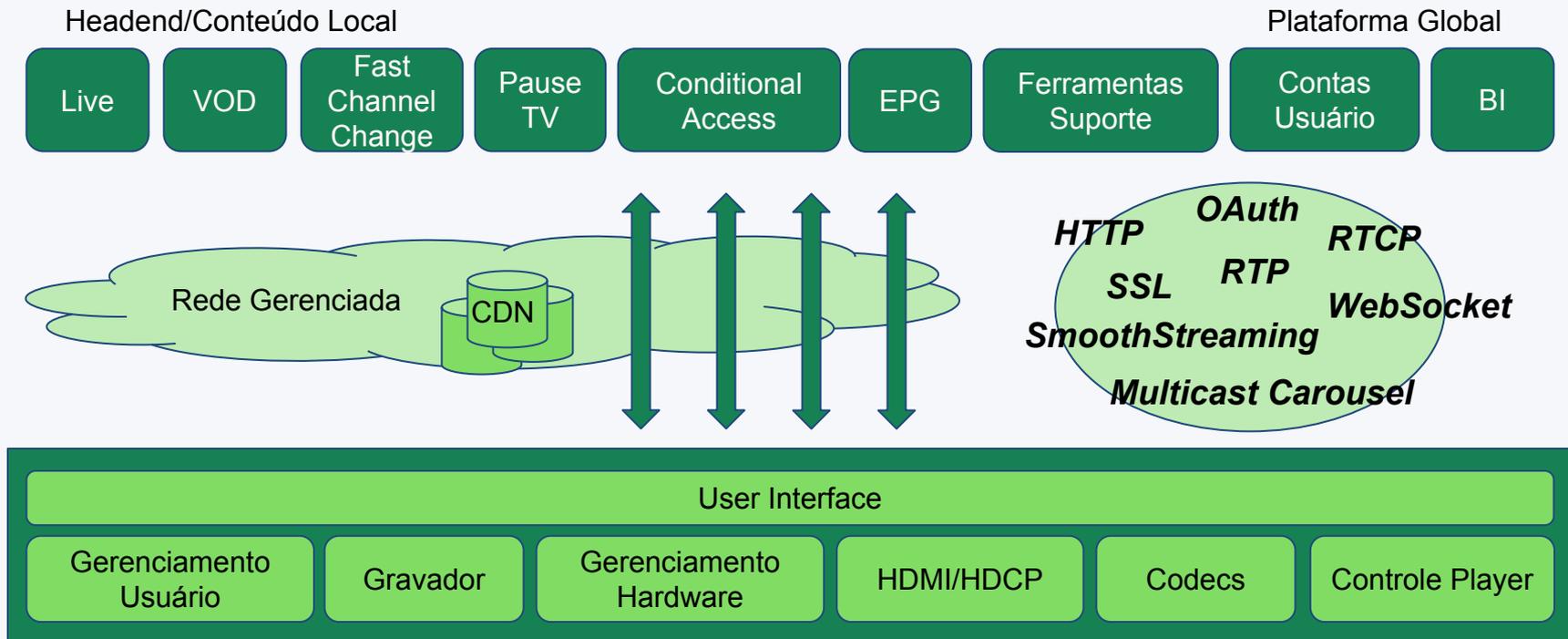


OpenPlatform Overview

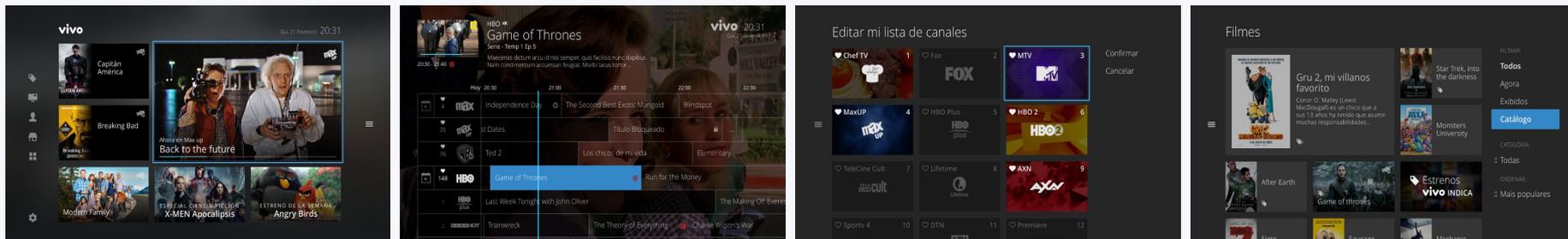
...



Overview Ambiente IPTV

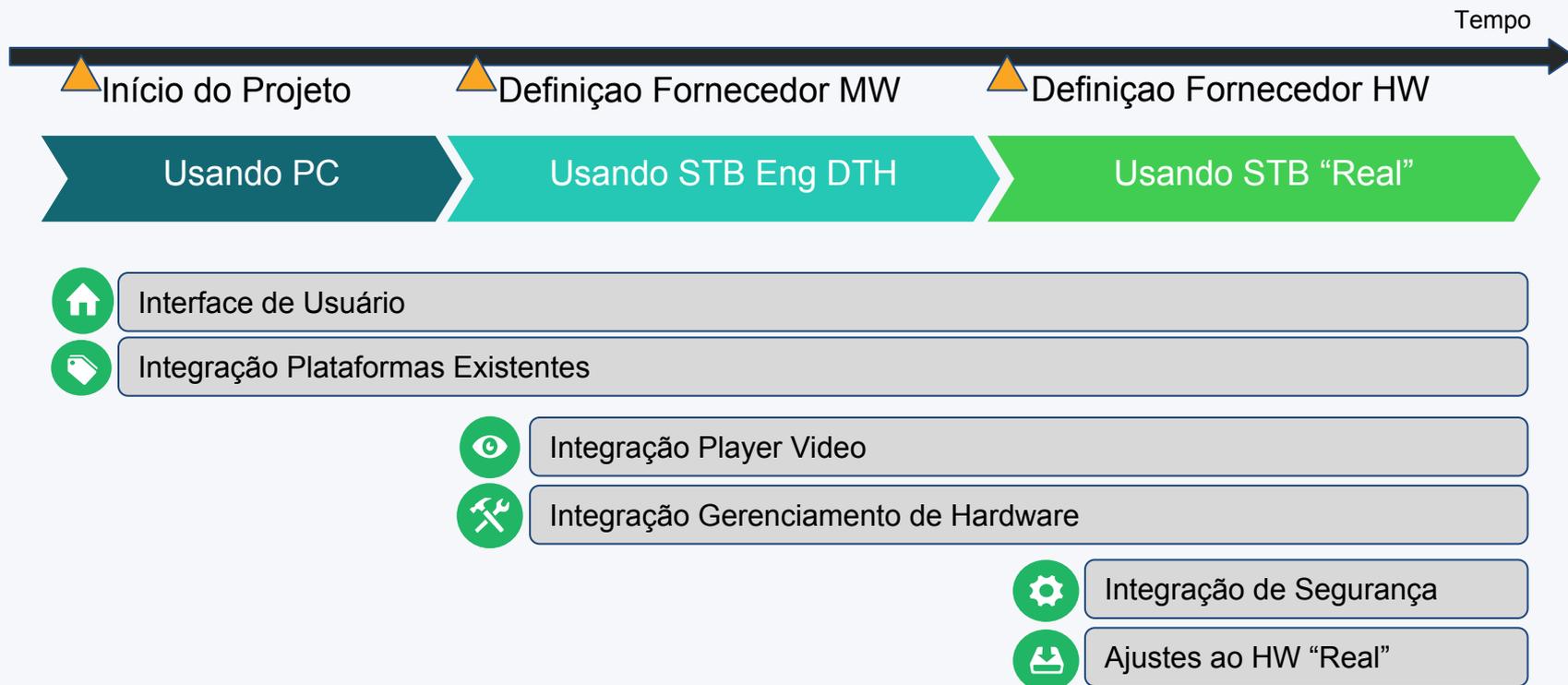


Escolhendo Qt



- ◆ Nova Interface de Usuário a ser definida por equipe de UX durante o desenvolvimento
- ◆ Quando iniciamos não tínhamos definição do HW a ser usado
 - Considerada a opção de migração ao STB legado (256MiB RAM, sem OpenGL)
- ◆ Decidimos usar Qt 4.8
 - Prova de conceito com STB Sigma 256MiB RAM satisfatório
 - Versão 4.8 suportada e fornecida por todos vendedores de chipset considerados
 - Melhor performance e menor footprint do que outras opções consideradas (Browsers HTML, Browser SVG proprietário)

Iniciando o Desenvolvimento

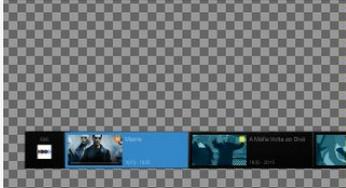


Como Usamos Qt?

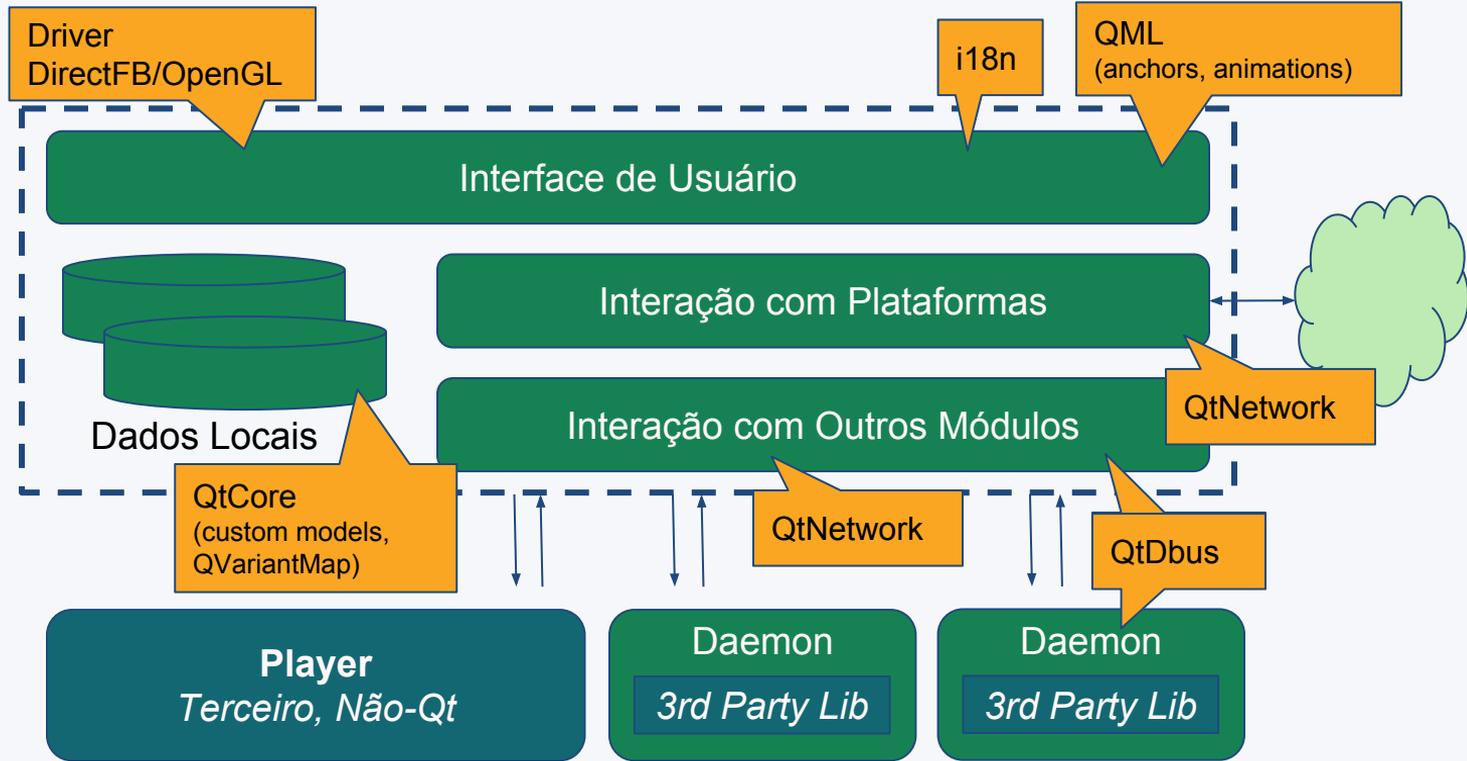
...



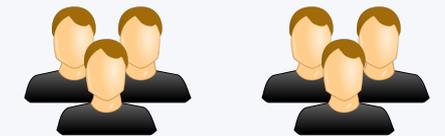
=



+



Transição Time Desenvolvimento



UX
(web+javascript)

Middleware
(C/C++)



Produto
(C/C++/QML)
20-25 devs

1

Unificação das equipes de desenvolvimento de UX e Middleware em uma única equipe responsável pelo produto como um todo

2

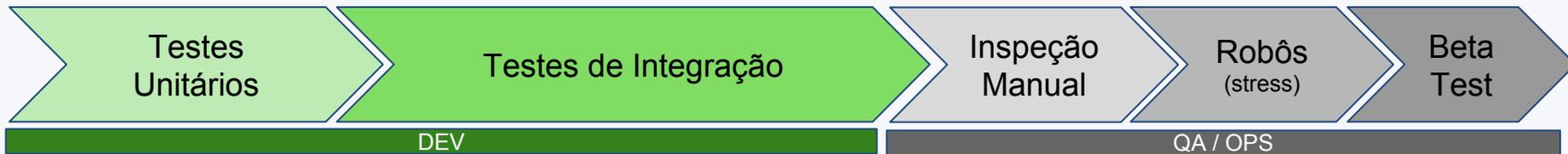
Melhor integração da equipe para solução de problemas

3

Desenvolvedores UX acostumados com JS+Angular migraram rapidamente para QML com feedback bastante positivo

- Maior produtividade, mais rápido fazer as coisas
- Possibilidade de otimizar o necessário em C/C++
- Performance muito melhor no STB
- Melhores ferramentas para debug e testes

Automação de Testes



- ◆ Qt Test
- ◆ Valida algoritmos complexos e máquinas de estado
- ◆ Executado em ambiente x86 após cada commit nos repositórios
- ◆ Não valida integração entre módulos

- ◆ Experimentamos várias abordagens
 - Automação baseada em GUI não foi efetiva
- ◆ Solução caseira, simples e efetiva:
 - Testes executam em um STB de engenharia com telnet habilitado
 - Valida comportamento através de logs “especiais”
 - Simula controle remoto com dbus-send + QApplication::sendEvents
 - Permite validar integração tanto com outros módulos do STB (player, controle de hardware, etc) quanto com plataforma (autenticação, permissões, etc)

- ◆ Test script em python
- ◆ Abre sessão telnet
- ◆ Envia comandos e compara logs (expect)

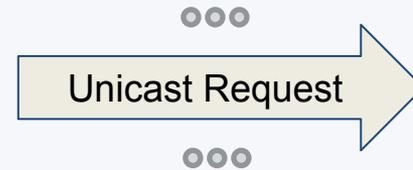
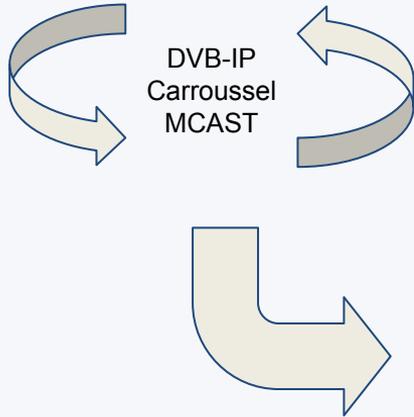


- ◆ Log de eventos
- ◆ Injeta QKeyEvent na aplicação ao receber evento D-Bus

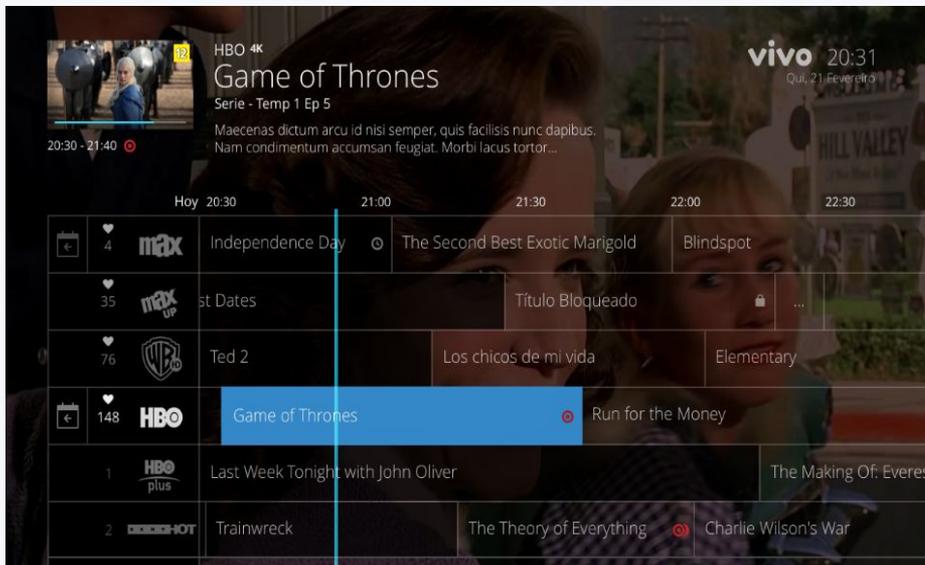
Qt Hints - Custom EPG Model

...

- ◆ Multicast up to a limit
- ◆ Unicast after that



Qt Hints - Custom EPG View



- ◆ Feito com muito malabarismo em QML
- ◆ Performance insuficiente para caixas legado
- ◆ Substituída por versão em C++ (QPainter, QDeclarativeItem::paint())

Qt Hints - Maskable Item



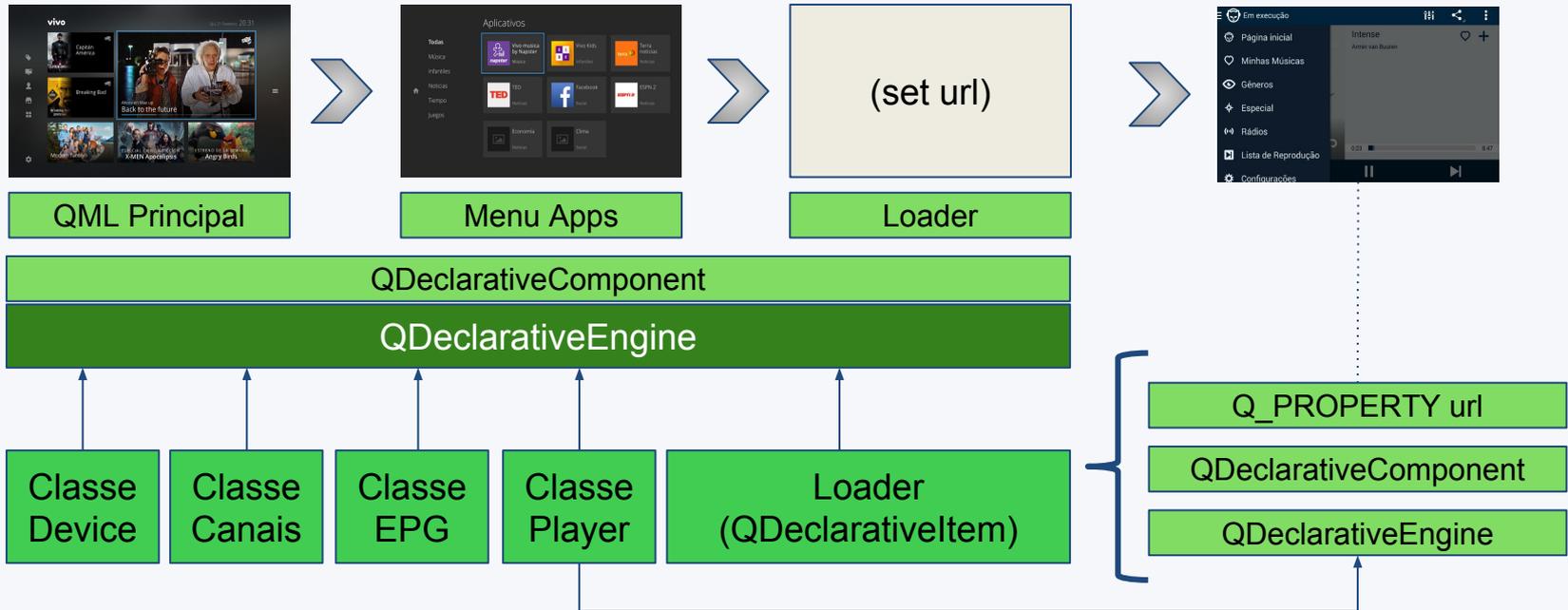
```
1. // QMaskableItem
2. QDeclarativeItem::setGraphicsEffect(QRectOpacityEffect())
3. // QRectOpacityEffect
4. QRegion totalArea = pixmap.rect()
5. QRegion renderArea = totalArea.xored(maskedArea)
6. painter.save()
7. // STB allows one clip for each draw operation
8. foreach (QRect rect , renderArea.rects()) {
9.     painter->setClipRegion(QRegion(rect));
10.    painter->setClipping(true);
11.    painter->setBackground(Qt::transparent);
12.    painter->drawPixmap(offset, pixmap);
13. }
14. painter.restore()
```



Qt Hints - Sandbox for Apps



- Usado para executar aplicações QML de terceiros com acesso limitado aos recursos
- Aplicação Vivo Musica desenvolvida em QML por uma empresa terceirizada



Qt Hints - Runtime Translation on QML



```
1. class Translator: public QObject
2. {
3.     Q_OBJECT
4.     Q_PROPERTY(QString language READ language WRITE setLanguage NOTIFY
languageChanged)
5.     Q_PROPERTY(QString tr READ tr NOTIFY languageChanged)
6. };
```

```
1. GphButton {
2.     id: exitButton
3.     text: qsTr("Quit") + tr.tr
4.     radius: 4
5. }
```



Qt Hints - Dynamic Menu



```
1. {
2.   "List":[ // List of Settings Group
3.     {
4.       "title":"Personal", // Title of Settings Group
5.       "widgets":[ // List of Settings Group Widgets
6.         {
7.           "title":"BLOQUEIO PARENTAL POR IDADE",
8.           "type":"PARENTAL", // Widget Type
9.           "options":[ { "text":"L", "value":0, ...} ]
10.        }
11.      ]

```

```
1. ListView {
2.   model: widgetsJson
3.   delegate: Item {
4.     property variant dynWdgt
5.     Column {
6.       id: widgetParent
7.       Component.onCompleted: {
8.         dynWdgt = Qt.createQmlObject("import QtQuick 1.1; " +
9.           widget.type, widgetParent, "dynWdgt" + index);
10.      }
11.    }

```



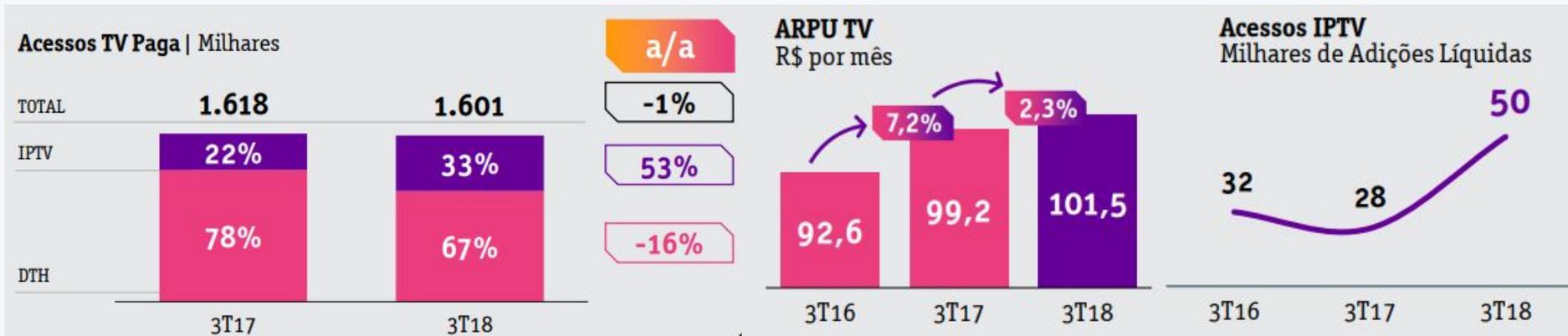
- ◆ JSON determina itens a serem carregados no menu central
- ◆ Vasta coleção de widgets

Resultados Comerciais Obtidos



Produto demonstra resultado comercial bastante positivo

- ◆ Produto já disponível em 98 cidades no Brasil
- ◆ Índice reclamação surpreendentemente baixo
- ◆ Feedback bastante positivo dos clientes
- ◆ Aumento no consumo de conteúdo, comparando com plataformas legadas
- ◆ Integração incremental de features (pip, netflix, recomendador, ver desde o início, etc)



Em Resumo



PERFORMANCE

Performance grafica excelente com footprint de memoria baixo



DESENVOLVEDORES

Aumento da produtividade dos desenvolvedores



TIME-TO-MARKET

Permitiu iniciar o desenvolvimento antes de ter todas as decisões tomadas



Agradecimentos

A organização do evento pela oportunidade de expor essa experiência, a Qt Company por facilitar nossas vidas e a comunidade por sempre estar disposta a ajudar.

Qt



Obrigado!

perguntas?

Milton Soares / Samuel Mello
milton.sfilho / samuel.mello @telefonica.com