

UI、图形和 EFL

Carsten Haitzler


韩国三星电子首席工程师

c.haitzler@samsung.com

Enlightenment / EFL 的创始人/负责人





The background is a dark, textured grey. On the left, there is a stylized illustration of a city skyline with various buildings and a prominent tower. Below the skyline are blue wavy lines representing water, with small green and white circles floating on the surface. In the top right corner, there is a cluster of colorful confetti in shades of blue, green, and purple. In the center-left, there are two overlapping dark grey circles, with a small yellow circle positioned between them.

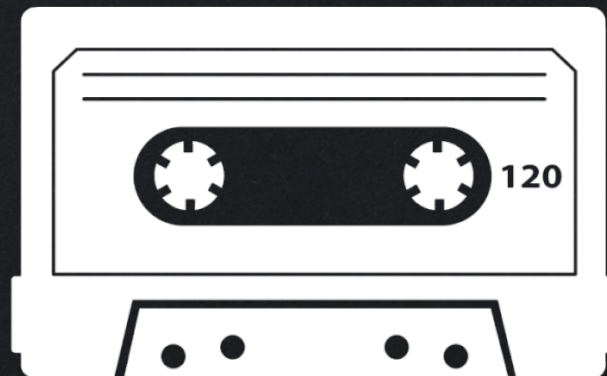
显示系统概述

图形

*Old
School*

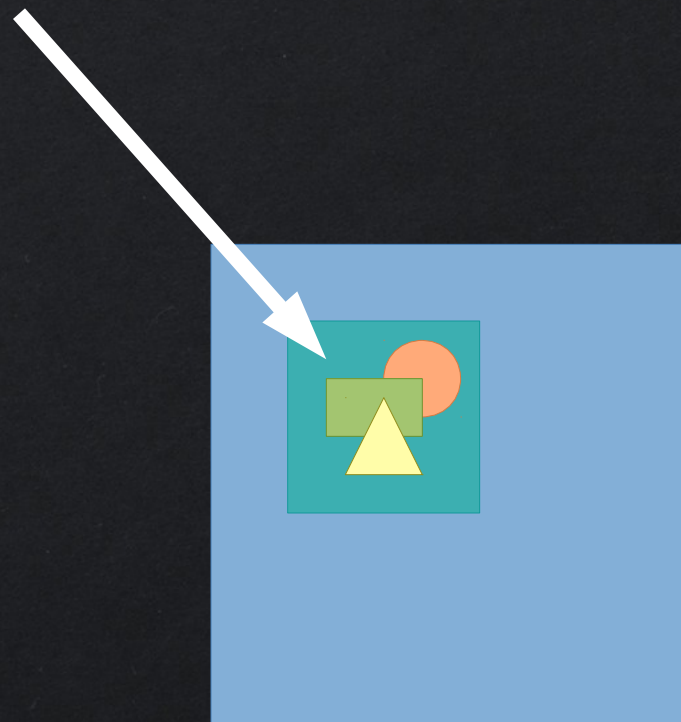
图形 – 老式 FB

- 过去我们直接使用帧缓冲器
- 如果您从事嵌入工作，就会对此很熟悉
- 功能手机大体上也采用这种方式运行
 - 应用程序“掌控”着屏幕（直接绘图）



单缓冲区 - 闪烁

CPU 或 2D 硬件生成
新像素/围绕 FB 复制

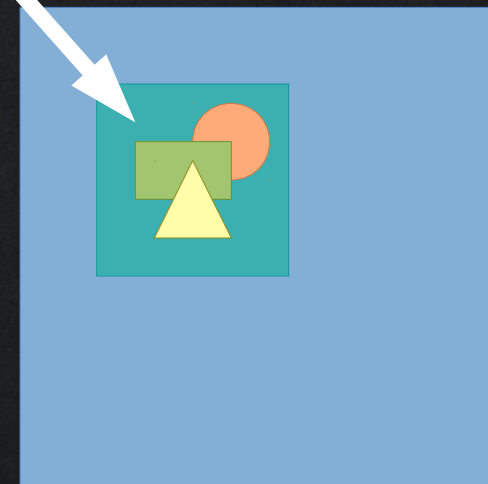


双缓冲区 - 复制区域

CPU 或 2D 硬件生成新像素

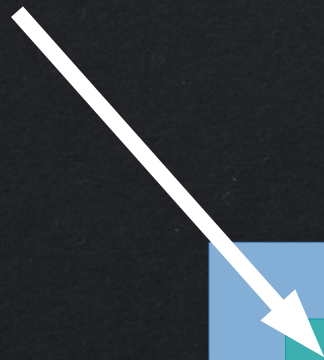


利用 CPU 或 2D 位块传送器硬件进行复制

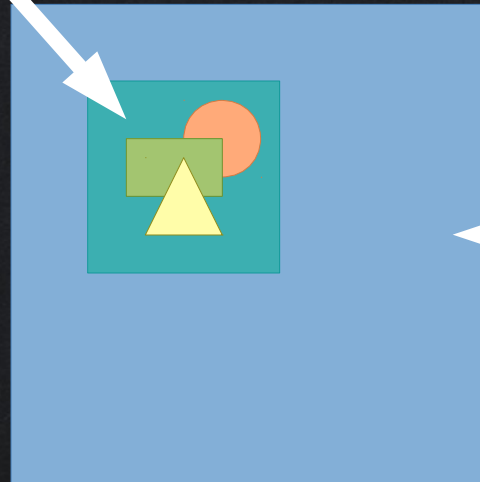


双缓冲区 - 交换缓冲区

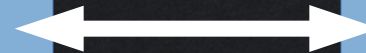
CPU 或 2D 硬件生成新像素



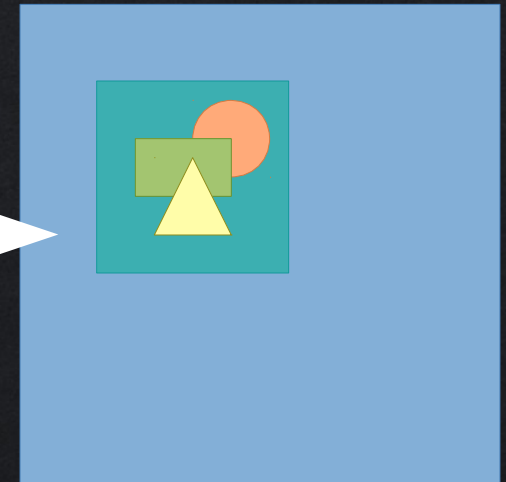
后台缓冲区



交换缓冲区



前台缓冲区



X11 – FB 共享（闪烁或复制）

- 构建于 20 世纪 80 年代，以支持共享 FB

—

— 还支持通过网络共享

— 支持通过网络的运行加速

- 一次可在屏幕上显示多个窗口

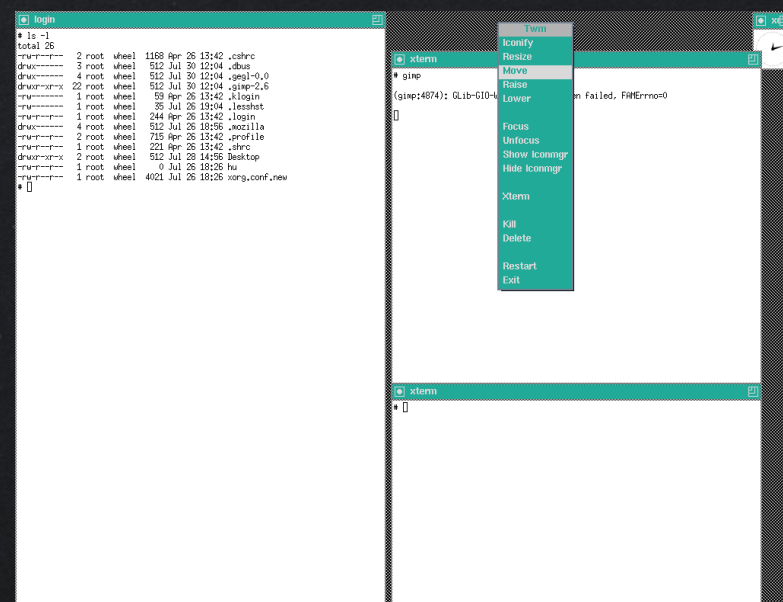
- 多屏幕

- 同时存在多个位深度

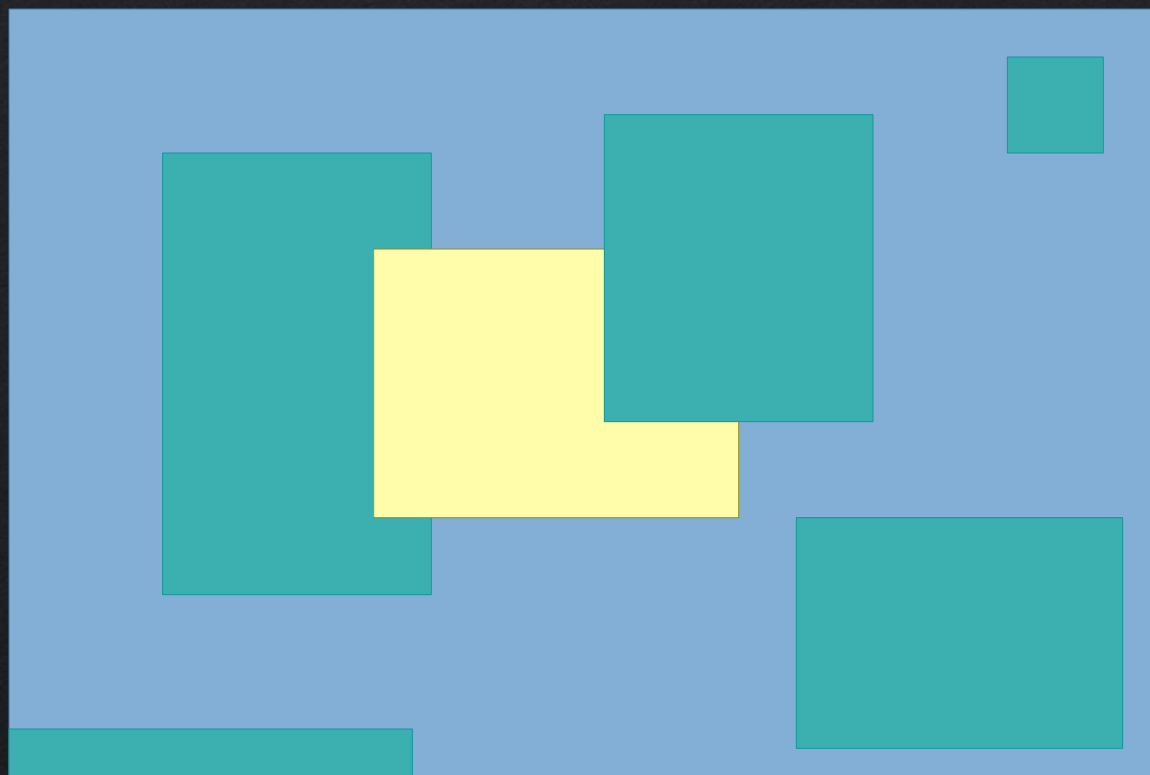
- 复杂

- 全部是矩形

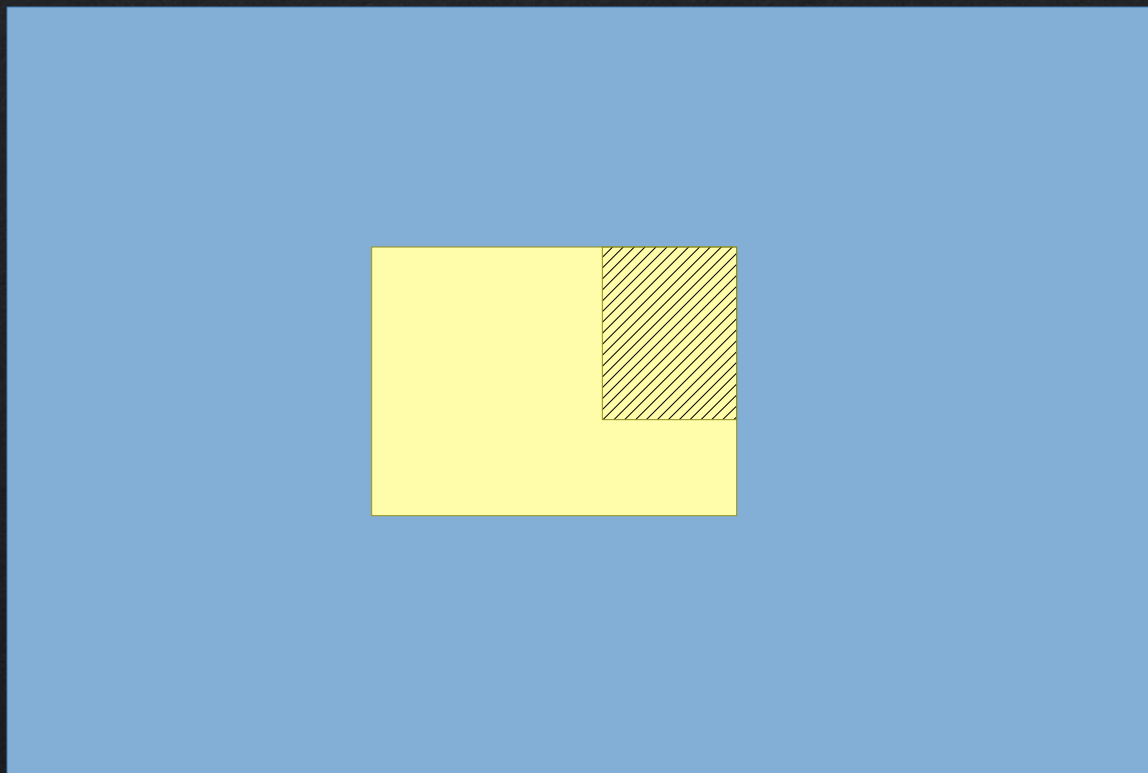
- 通过请求绘制服务器端



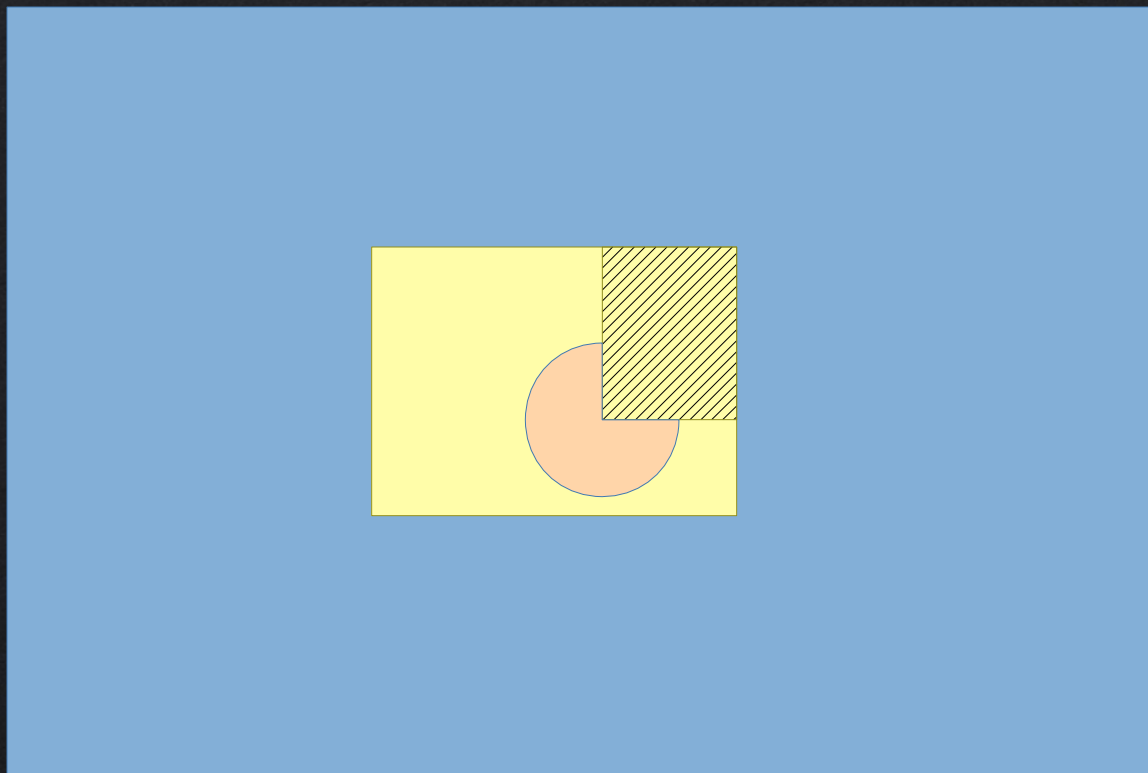
X11 绘图



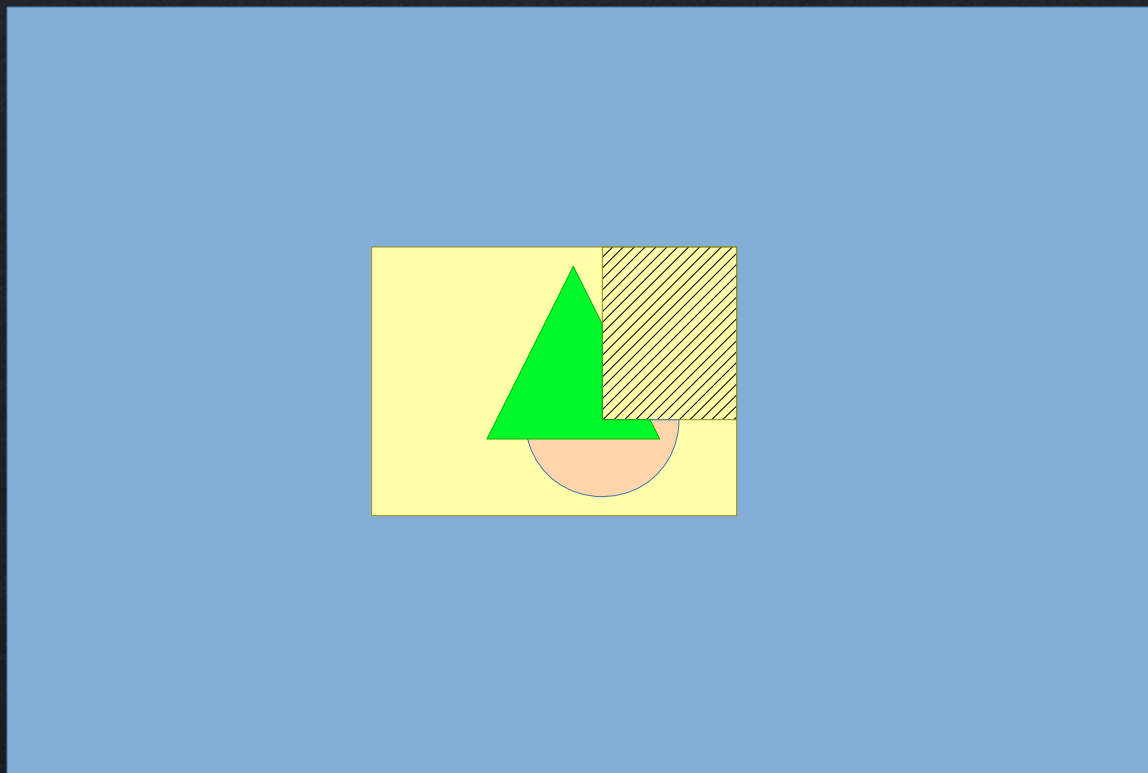
X11 绘图



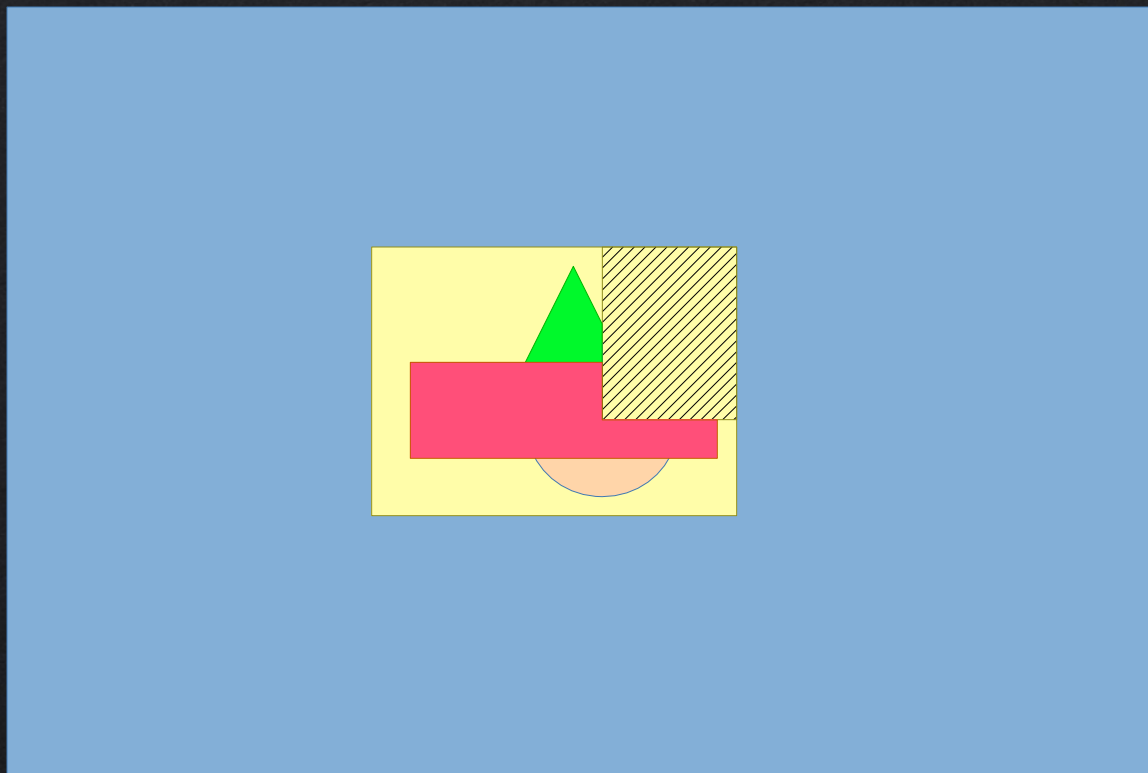
X11 绘图



X11 绘图



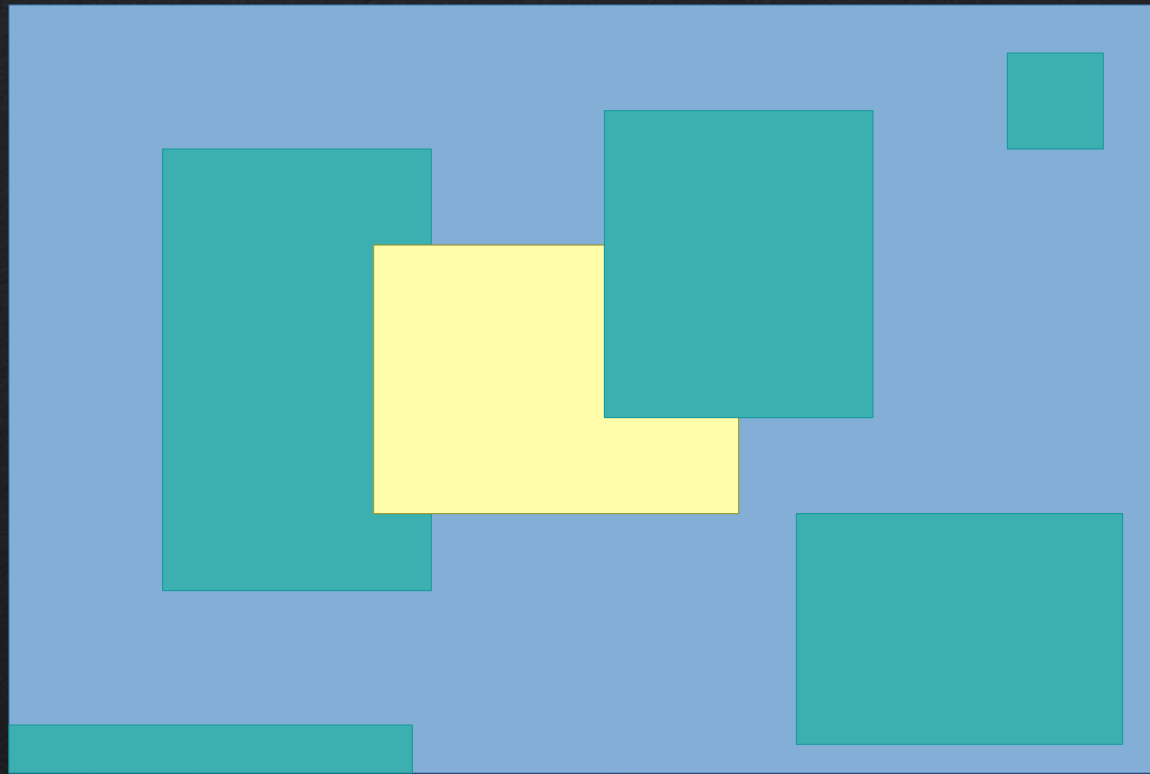
X11 绘图



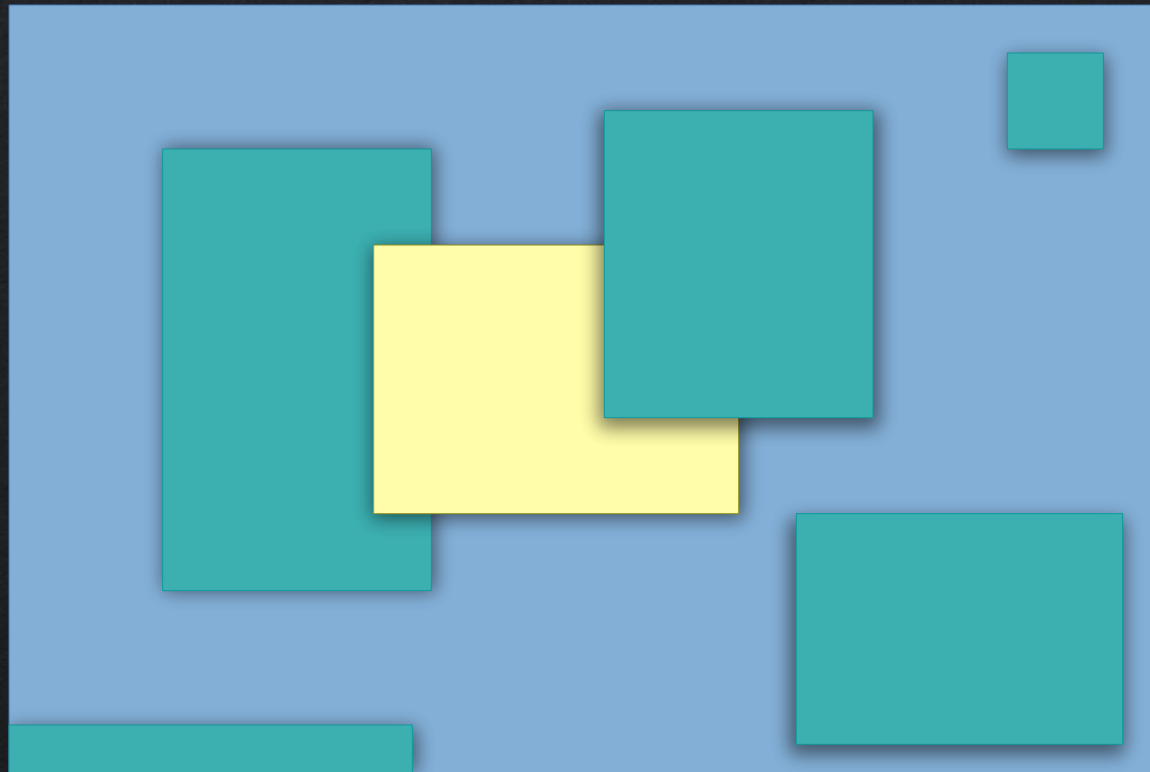
X11 添加影像合成

- 通过若干扩展添加影像合成
 - 合成、损坏、修复
- 强制执行渲染以进入每个窗口屏幕外的象图
- 允许合成器捕获正在变化的事件和象图 ID
- 这可支持合成器添加阴影、缩放等特效
- 下降趋势 – 不会影响事件（直接进入目标客户机）

影像合成



影像合成



其他操作系统也采用相同方式

- **Windows XP** 到**Vista+** 添加的影像合成
- **Mac OS9** 到**OS X** 添加的影像合成
- 影像合成在所有这些操作系统中均采用相同的核心理念
 - 所有到窗口的绘图现在都进入后台缓冲区
 - **合成器**能够访问后台缓冲区和更新
 - **合成器流程**使用缓冲区合成屏幕图像
 - 该合成过程可添加特效和透明度
- **Tizen** 也支持影像合成！

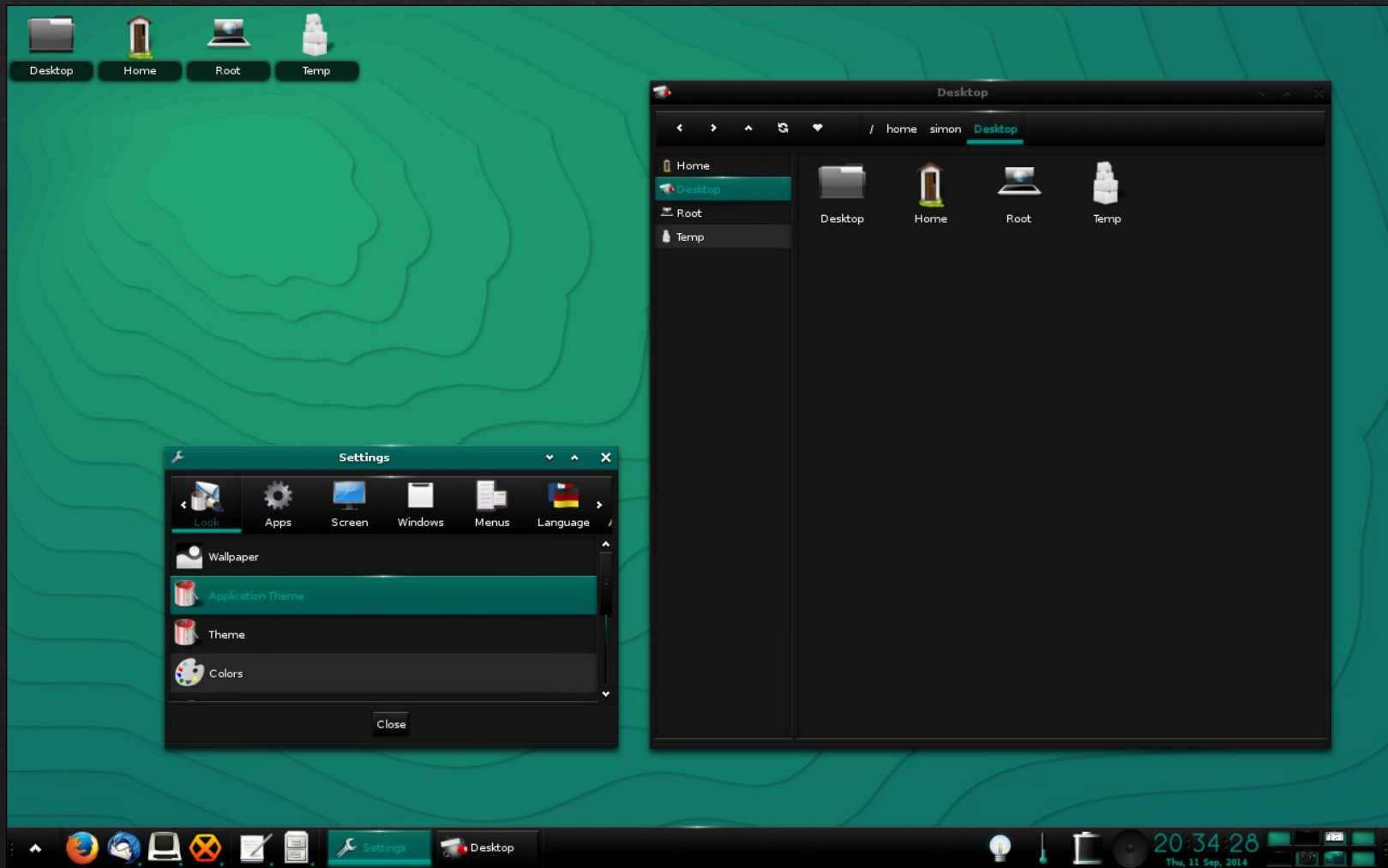
Tizen 2.x 显示

- **X11 + 合成器(Enlightenment 0.17)**
 - IVI 除外 (**Weston/Wayland**)
- 它是一个位于底层的全桌面 **WM + 合成器**
 - 窗口能够（调整大小）
 - 窗口能够移动
 - 窗口可以不占满整个屏幕
 - 可以同时打开多个窗口
- 只有一些策略（如移动）强制要求简化事物
 - 在大多数情况下，通常不会调整屏幕大小
 - 窗口往往也不会移动

Tizen 2.x WM



Tizen 2.x WM



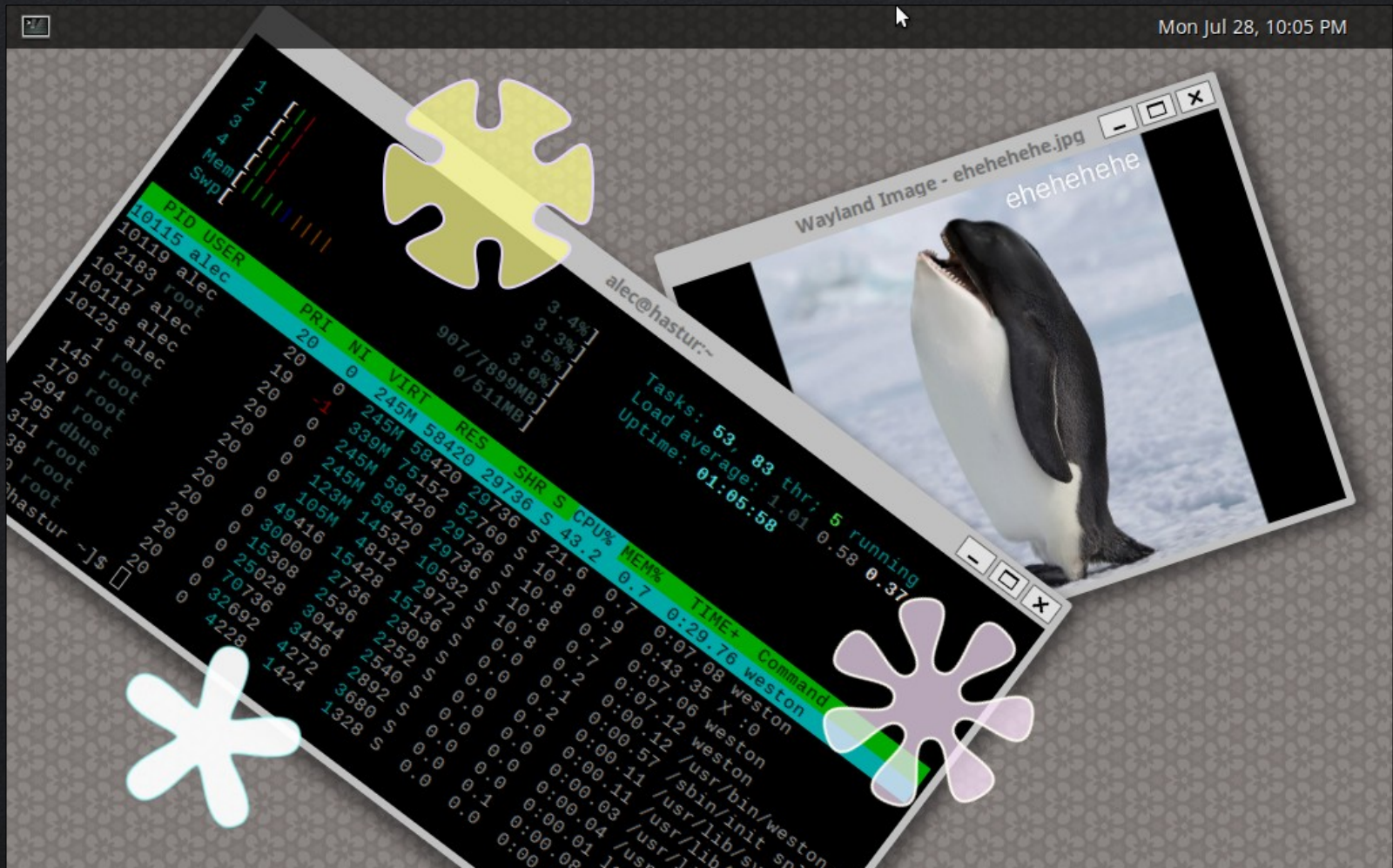
Tizen 3.x 显示

- **X11** 和 **Wayland** 均受支持
 - 迁移至 **Wayland**，弃用 **X11**
 - 不要假设/期望或使用任何 **X11**
 - 存在可隐藏 **X11** 的抽象 – 请好好加以利用
 - **IVI** 除外 – 仅支持 **Wayland**

Wayland

- 比 **X11** 更简单
- 所有位置都是一个缓冲区（或平面）而不是矩形
- 仅需要合成显示
 - 专为支持快速路径零复制交换而设计
 - 全屏应用程序（如果存在多个硬件层，则有多个窗口）
- **Weston** 是当前的演示合成器
 - 更多桌面作为合成器添加支持
 - **GNOME、KDE、Enlightenment ...**
- 主要的工具套件现在都已有良好的 **Wayland** 支持
 - **EFL、Qt、GTK+、SDL、...**

Wayland



为何选择 Wayland

- 安全（**X11** 从设计开始就存在**不安全性**）
- 在**单个流程**中发挥合成器 + **WM** + 显示服务器的作用
 - 轻松支持输入转换
 - 比多个流程更加高效
 - **更低的功耗**
- 合成模式更加整洁，支持**硬件层**使用
 - 更轻松支持 **YUV** 或 **RGBA** 层（底层）
- **精简专家**
 - 淘汰了传统的服务器端渲染
 - 目前 **X11** 还支持客户机进行**自渲染**

为何选择 Wayland

- 输入法支持是内置支持，而非事后添加
- DND 为内置而非后来添加
- 比 X11 提供更多客户机隔离
- 需支持的代码减少
- 新 GPU 和主板上市速度加快
- 基于 DRM、KMS 等开放标准构建
- 确保“每一帧皆是完美的”的几率提高
- 对于命令+信号而言仍然是客户机-服务器（UNIX 插槽）
 - 缓冲区为零复制（仅支持通过 IPC 发送的处理）

The background is a dark, textured grey. On the left, there is a stylized illustration of a city skyline with various buildings and a prominent tower. Below the skyline are blue wavy lines representing water, with small colorful circles floating in it. In the top right corner, there is a cluster of colorful confetti in shades of blue, green, and purple. In the center-left, there are two overlapping dark grey circles, with a small yellow circle positioned between them.

应用工具套件

为第一方构建应用程序

- 能够访问任何层级（**X11**、**Wayland**、**FB** 等）
 - 考虑了可移植性以及从 **X11** 到 **Wayland** 的迁移
- 能够直接使用 **OpenGL**
- 能够直接使用 **EFL**
- 能够直接使用 **Qt**
- 等

为第三方构建应用程序

- 使用**HTML5 + Web** 运行
 - 提供 **HTML5 DOM / CSS / JS**
 - 强制大多数应用程序采用 **JS** 格式
 - 启动速度慢且占用内存空间大
 - 为了开发速度/环境需要牺牲性能

为第三方构建应用程序

- C++ Tizen::原生 API 被弃用

为第三方构建应用程序

- 全新 Tizen 原生 C API
 - 最新版本为 Tizen 2.3
 - 大量 API 覆盖 Tizen 设备 UI 的各个方面
 - API 是 EFL
 - EFL 1.7 + 补丁（仅支持一些 EFL 库）
- C 语言是核心支持
 - 当然同样可支持 C++



EFL

什么是 EFL ？

- 代表： Enlightenment Foundation Libraries
- 作为构建 Enlightenment 的一部分而创建
 - <http://www.enlightenment.org>
 - “我们需要这些支持，其他工具无法提供”
 - “或许它们对其他内容也非常有用？”
 - 采用库而非内置代码的形式
- 构建时考虑了“嵌入”
- 只需由一个小型重点团队创建和维护
- **100%** 开源（开发模型、社区和代码）

什么是EFL ?

- 现在最多大约有**1,000,000** 行 **C** (1.11)
- 是一个以 **C** 为中心的采用 **C API** 的库
- 包含带有专用名称、函数和层级的子库
 - **Elementary** – 高级 **API** + 小工具 **Eva** – 核心场景图 + 渲染
 - **Ecore** – **Mainloop**、事件、消息和动画/计时
 - **Eina** – 数据结构和低层级
 - **Edje** – 来自磁盘主题的 **Canvas** 对象“元”文件
 - ...以及其他

分区！

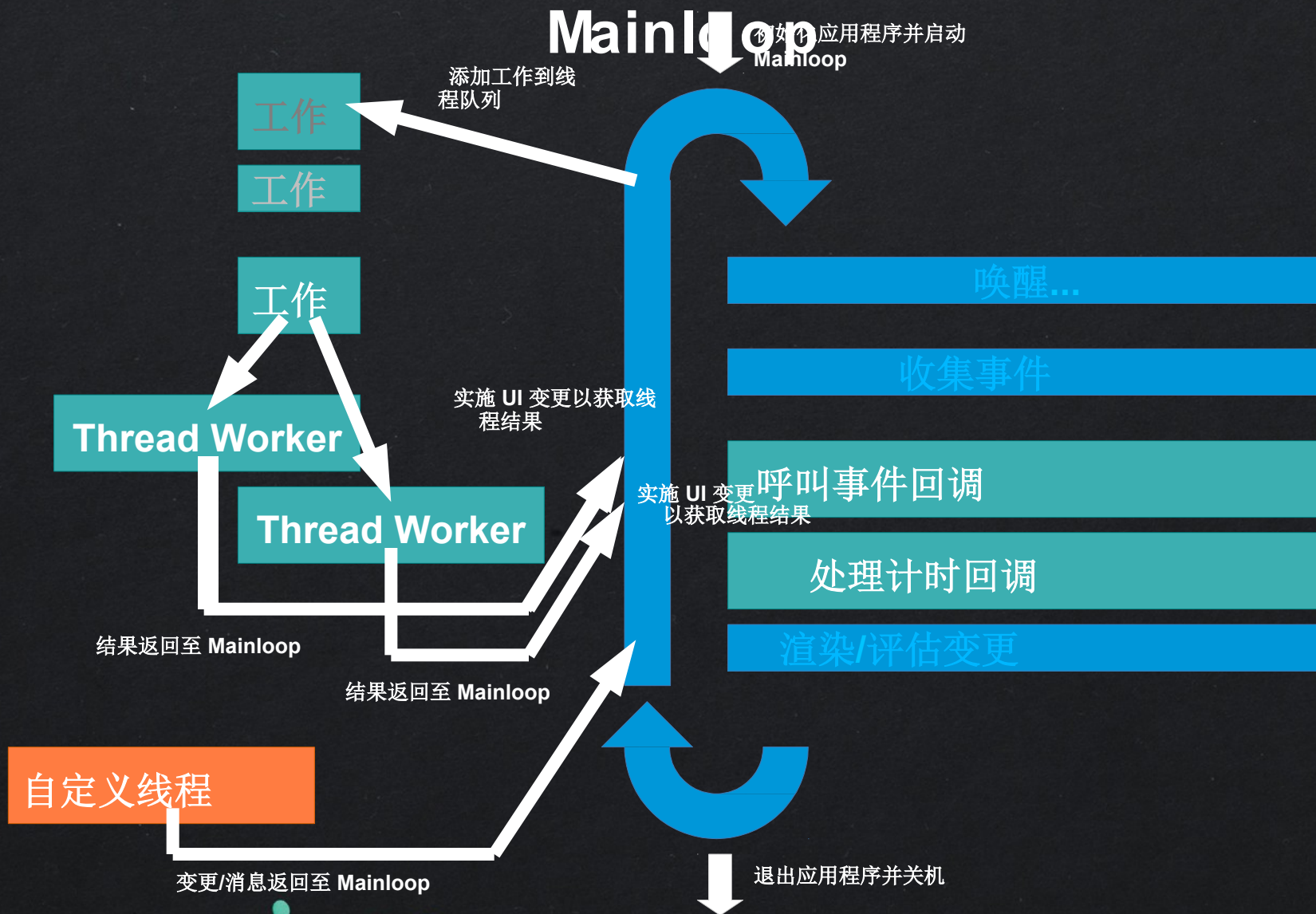
HTML5 应用程序



*大体分区图 仅供参考

Mainloop 模式

- EFL 旨在实现以**Mainloop** 为中心
- **Mainloop** 在一个线程中处理应用**状态**和 **UI**
- 子线程能够异步执行任务并将信息传送给 **mainloop**
- 为您提供 **Thread worker** 池 **API**
- 鼓励数据/任务的线程**隔离**
- 鼓励当前状态到 **Mainloop** 的串行连接
 - 状态变化的隐式同步
 - 所需的系统锁减少
 - 线程漏洞减少



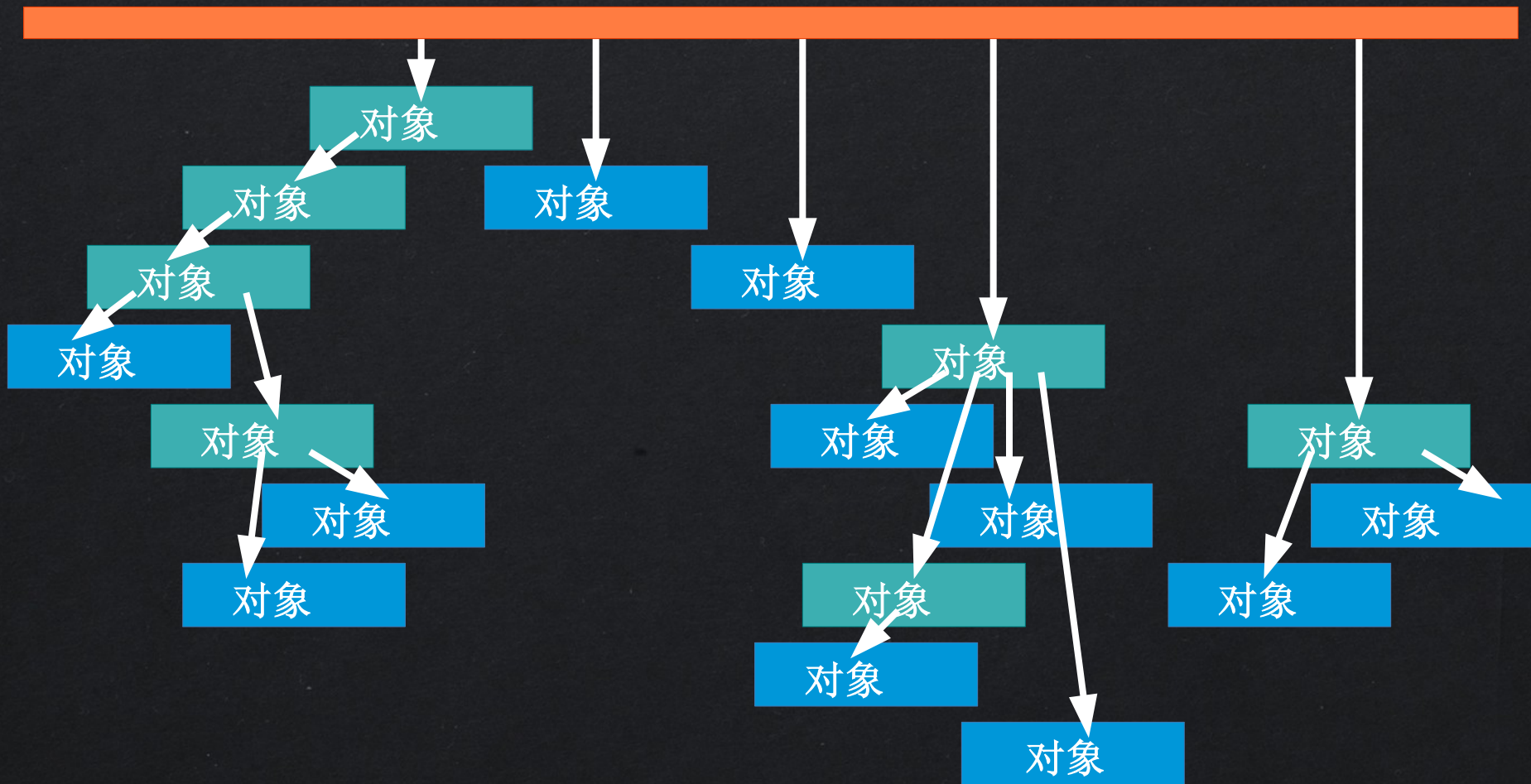
小工具

- 按钮
- 滚动条
- 条目
- 复选按钮
- 单选按钮
- 列表
- 文本框、表格、网
- 格、菜单
- 工具栏
- 以及更多

场景图

- 几乎不同于所有位置，这里没有“渲染” API
- 但是存在场景图（保留的模式渲染）
- 场景图按照对象描述窗口内容
 - 每个对象都是原始对象
 - 文本、图像、矩形、容器...
 - 按钮、表格、文本框、列表...
 - 不重复绘图。修改对象以获得变更场景
 - 场景图计算如何重复绘图 – 而非应用场景
 - 场景图计算应进行哪些变更以最大限度减少工作
 - 更多对象意味着需要更多用于计算变更的成本:(

场景图



场景图

- 每个对象都拥有几何结构
- 每个对象都拥有堆栈（在其他对象上面/下面）
- 一些对象（智能对象 – 容器）可拥有子对象
- 每个顶层的对象（非母对象）位于一个层级中 层级
- 具有绝对的堆栈优先权（0 在 1 之下，1 在 2 之下）
- 它不仅支持空间排列还支持 Z 序排列
- Z 序（堆栈）对于获得一些特效/行为而言非常关键
- 由于其自身属性，对象是合成而来，并非“继承”而来

我没说实话 – 您可以渲染

- 对于向量，可使用 **Cairo + Evas** 映像对象
 - 基于映像对象像素数据包裹 **Cairo Surface**
 - 绘制到 **Cairo Surface**（可在线程中执行）
 - 如果在线程中执行，请使用双缓冲区映像对象
 - 抛弃 **Cairo Surface**
 - 恢复像素数据设置并为 **evas** 区域提供映像更新
- 对于任何自定义像素数据，您可以使用同样的方法
 - 映像对象是 **ARGB8888** 像素

我没说实话 – 您可以渲染 (OpenGL)

- 您可以将 **OpenGL** 渲染“插入”到场景图
- 使用 **Elm GLView** 小工具可以节省时间
- 由于是场景图，它会强制执行一些限制
 - 必须使用 **Evas GL** 界面和背景处理
 - 这可支持您的 **GL** 渲染实现零复制渲染
 - 不要超过原始渲染（无复制）
 - 支持您的 **GL** 拥有控制权并叠加在对象上
 - 支持 **canvas** 对象覆盖您的 **GL** 渲染
- 使添加文本、小工具、**HUD** 和调试游戏等更简单
- 提高了 **GL** 的可移植性（所有平台上的 **GLES2**）

未来

- 已经到来 (**EFL 1.11** 已经上市)
 - 很遗憾, **Tizen** 晚于 **EFL** 推出 (**Tizen 2.x**)
 - **Tizen 3.x** 正在追赶上游 **EFL**
- 上游 **EFL** 进行了大量改进并添加了新特性
 - 用于保障安全的 **EO (object infra)** 以及 **C Evas** 中的 **OO**
 - **3D** 对象
 - 文本过滤器 (接下来也将支持图像)
 - 优化
 - 清理
 - 更出色的 **Windows / Mac** 支持 (移植) ...

未来

- 视频对象...
- **C++ API** （自动生成– 当前“不稳定”）
- **LUA** 应用程序创建 + 自动生成 **API**
- 相同的 **Python API** （自动生成）
- 以及更多...<http://www.enlightenment.org>
- <http://git.enlightenment.org>
- <http://phab.enlightenment.org>
-



问答

有问必答



TIZENTM

DEVELOPER SUMMIT 2014



SHANGHAI



TIZEN开发者峰会（上海）