

嵌入式工业计算机和人机界面的软件创新

Embedded industrial computers and HMI software innovation

何小庆 Allan he

中国软件行业协会嵌入式系统分会副秘书长



发言内容 (Content)

1. 嵌入式系统和IPC的历史

2. 嵌入式与 IPC的发展前景

3. 图形用户界面的技术

4. Android UI定制和实例

5. 未来UI软件技术发展趋势

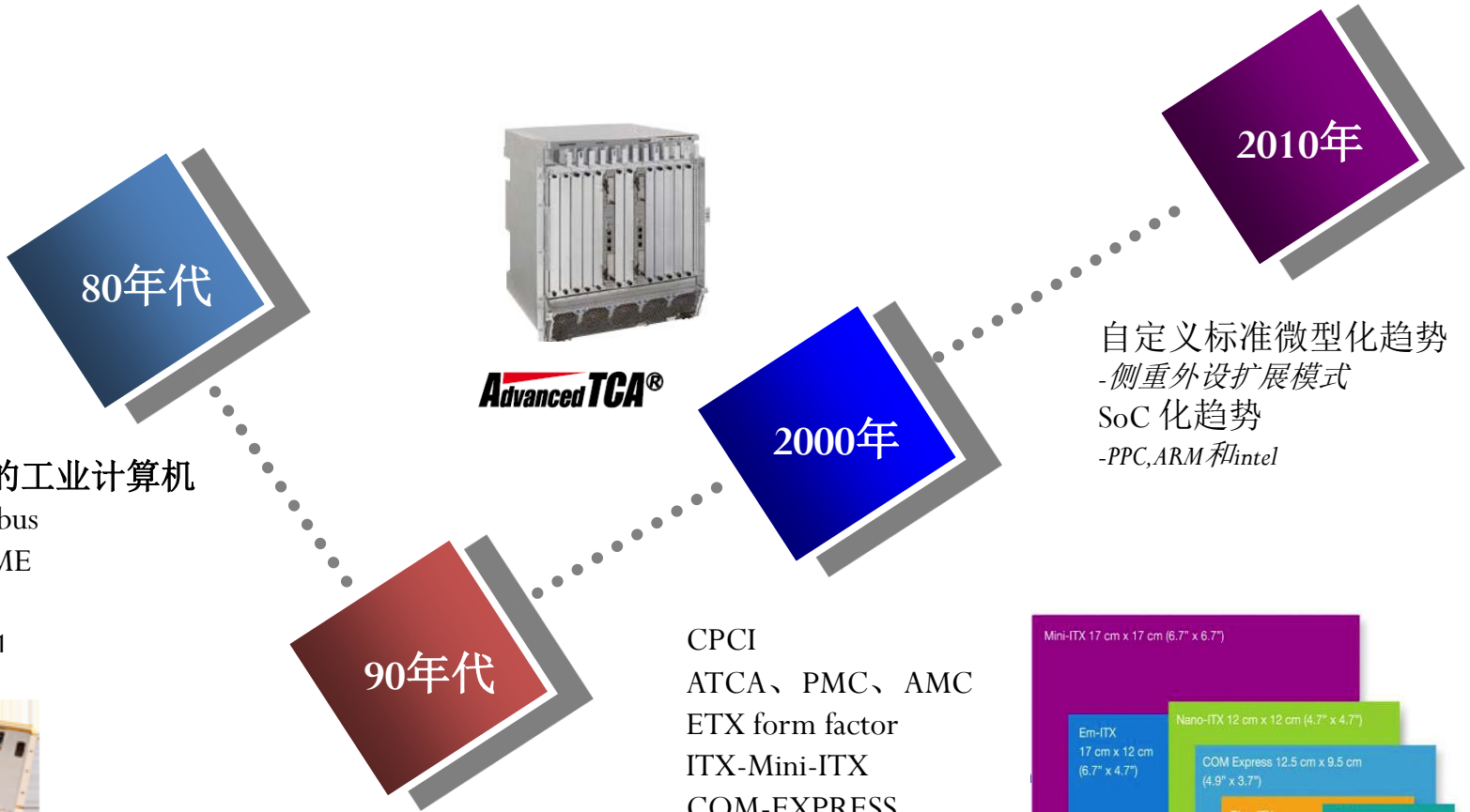
嵌入式系统的起源

- 嵌入式系统与微处理器密切相关。
 - 70年代微处理器问世。
 - 80年代MCU、DSP 出现。
 - 90年代是百花齐放时代。
 - 2000年以后ARM 开始成为主流，主导移动终端市场。
 - 2010年以后ARM 渐成MCU主流并进军服务器市场。
- 嵌入式智能是大趋势。
 - 微处理器之前，产品里面加入计算机是不可思议的事情，今天任何的电子产品如果没有智能，好像就不太正常。

Before the Microprocessor ,it was absurd to consider adding a computer to a product ;now in general, only the quirky build anything electronic without embedded intelligence.

- Jack G.Gannles ESD 杂志

工业计算机发展历史



80年代

专用总线的工业计算机

- Intel Multibus
- Motorola VME
- Zilog STD
- DEC-PDP11



VMEbus
Technology



PC/104



Advanced TCA®

2000年

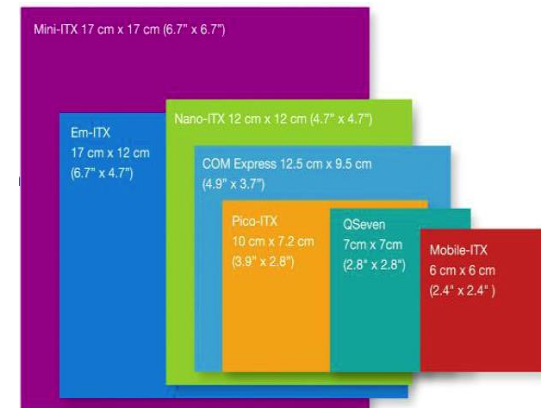
- CPCI
- ATCA、PMC、AMC
- ETX form factor
- ITX-Mini-ITX
- COM-EXPRESS
- PC-104+



COM Express®

2010年

自定义标准微型化趋势
-侧重外设扩展模式
SoC 化趋势
-PPC, ARM 和 intel



嵌入式软件的发展历史

80年代 → 90年代 → 2000年 → 2010年 → 2020年

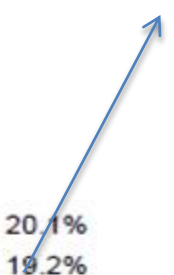


嵌入式OS发展历史

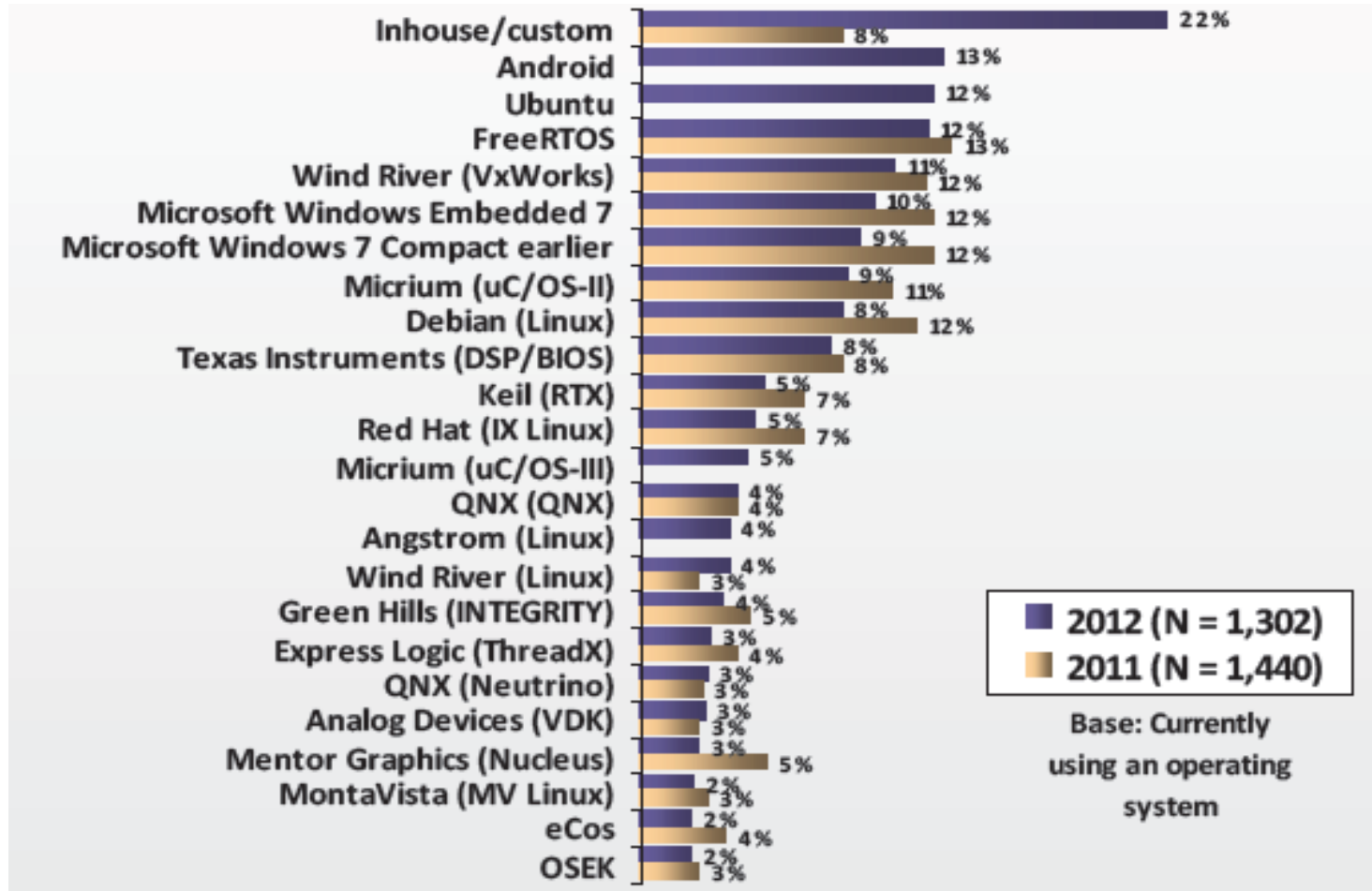
- 80年代商用嵌入式RTOS出现。
 - OS-9、VRTX、Psos、Vxwork和QNX。
- 90年代通用OS进入嵌入式系统。
 - Windows Embedded/CE, Linux(Montavista和Windriver ...)
- 2010年以后是Android广泛流行。
 - Android 和Linux ,ARM 已经占了先机。

| 2009 | | 2010 | | 2011 | |
|-----------------------|-------|-----------------------|-------|-----------------------|-------|
| In-House | 26.9% | In-House | 20.4% | In-House | 20.1% |
| Microsoft Windows CE | 26.3% | Microsoft Windows CE | 19.1% | Android (OS) | 19.2% |
| Microsoft XP Embedded | 25.4% | Red Hat | 18.3% | Microsoft XP Embedded | 16.4% |
| Microsoft - Other | 21.1% | Microsoft XP Embedded | 17.1% | Microsoft CE | 15.9% |
| Microsoft DOS | 18.4% | Microsoft DOS | 12.2% | Red Hat | 13.4% |
| Red Hat | 17.0% | Wind River VxWorks 5 | 11.3% | Microsoft DOS | 12.3% |
| DOS (other than MS) | 13.5% | Microsoft - Other | 9.7% | Microsoft Other | 11.8% |
| LynxWorks LynxOS | 9.4% | Other embedded Linux | 9.7% | Wind River VxWorks 6 | 8.6% |
| Wind River VxWorks 5 | 8.5% | <u>Android</u> | 9.4% | Wind River VxWorks 5 | 7.8% |
| Green Hills INTEGRITY | 7.0% | DOS (other than MS) | 8.8% | LynxOS | 6.2% |
| Other embedded Linux | 7.0% | Micrium uC/OS-II | 7.4% | QNX | 6.2% |
| Other | 7.0% | Wind River VxWorks 6 | 7.2% | Micrium | 5.8% |

增加50%



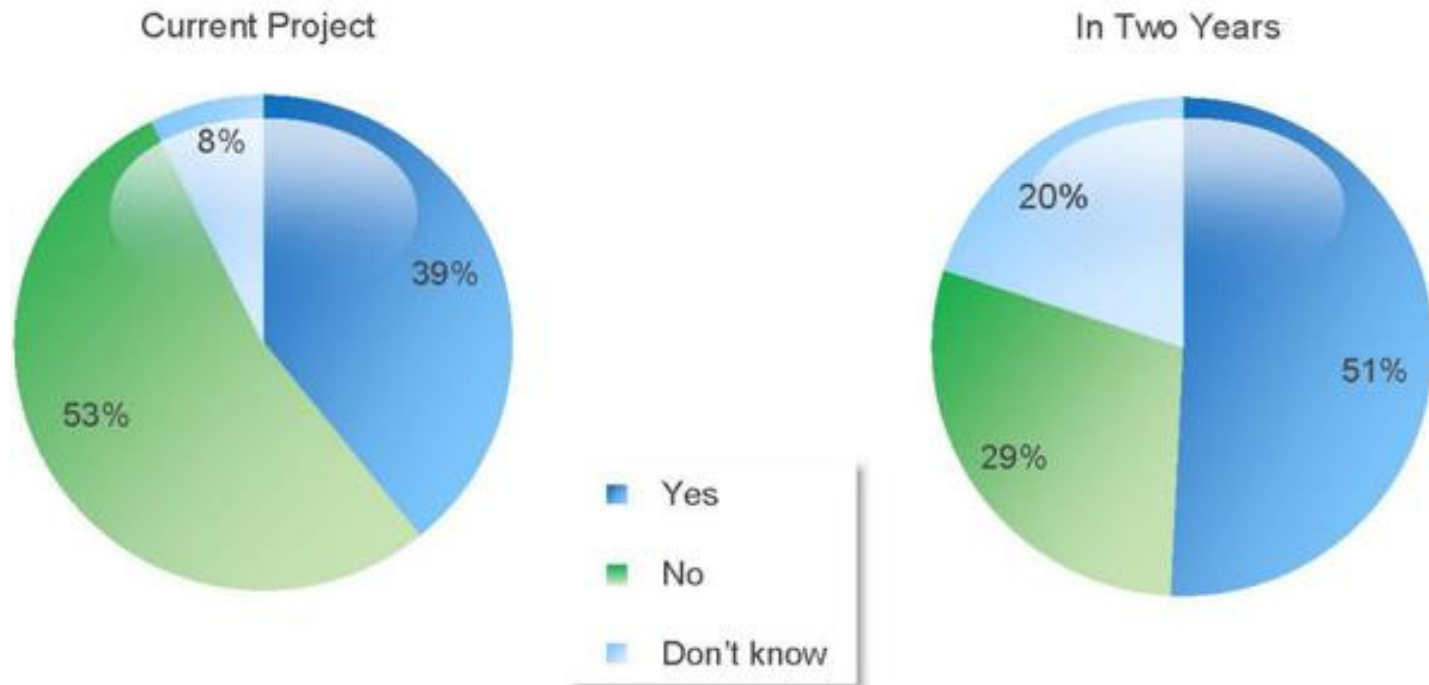
正在使用的嵌入式OS情况-2012年



未来：互联网、无线和云计算

- 嵌入式系统设计需要考虑无线互联和服务模式改变的影响。计算无处不在：云端、手机、平板、PC和各种嵌入式设备。

Inclusion of Web Component (e.g., UI, or Web Services)
in Current Embedded Project & Expected on a Similar Project in Two Years
(Percent of Respondents)



未来：嵌入式产品变的好看和时尚



未来：安全还是关键

Railway: **EN 50128 SWSIL 4**,
Security: **EAL6+ High Robustness**
Medical: **FDA Class III**, app
Industrial: **IEC 61508 SIL 3**,
Avionics: **DO-178B Level A**

• 测试和认证

- 航电软件规模越来越大：F-22的航电系统有170万行源代码，F-35则激增到600万行源代码。安全关键、任务关键和高可信，严格认证符合DO-178B A软件要求的代码：每行代码100-150美元；
- 汽车电子软件日益复杂，从1K-10K发动机、1M-100M行代码的ABS到IVT车载信息系统，认证和测试工作量越来越大。

• 安全软件

- 医疗电子、无线传感网、移动终端和汽车电子等关键工业应用的身份认证和防范黑客入侵。

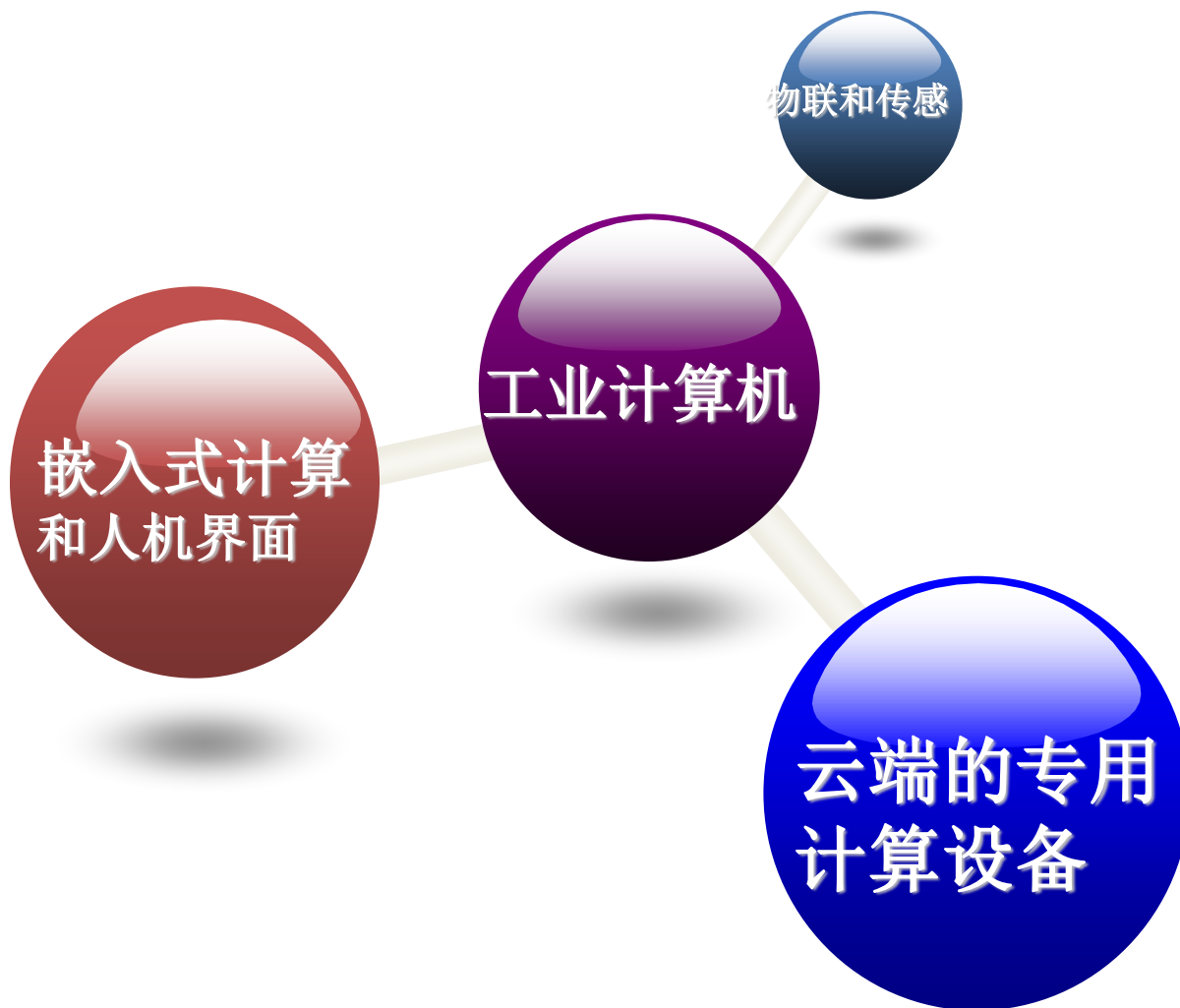


未来：软件变得更重要

- RTOS 主宰的嵌入式系统正在改变。
 - 强实时、高可靠和微型装置的系统依然会采用，但是其他的系统将转到开源的Linux（比如 Android）；基本系统的硬件移植和驱动开发将由芯片公司完成。
- 嵌入式开发最终将跨越C编程时代。
 - C++、Java 和web编程渐成主流。
- 芯片的选择将由行业的应用和软件主导和驱动。
 - Android 成为移动标准，Autosar 成为ECU标准
- 嵌入式软件开发持续增温。
 - 进入90时代软件代码急剧增加，工作量大大超过硬件
 - 软件帮助产品的差异化需求、降低研发和维护成本。



工业计算机（IPC）发展趋势



工业计算机（IPC）发展的关键技术

工业计算机 发展的关键技术

小型化-微型化-嵌入化

无风扇-节能-低功耗

一体化-高集成-SoC

高性能计算-多媒体数据处理和计算

专用实时OS-通用OS-嵌入式OS

人机界面发展历史

- CRT显示、图形监视器（墙）、单色LCD、伪彩色、TFT真彩、LCD显示（墙）、触摸和多点触摸屏。
- 今天的人机界面多数都是图形用户界面（GUI），其他的人机界面方式，比如手势、体态、语音和视觉等都还处在小规模应用和计算机科学研究范围。
- 多点触摸屏是一种新型的、很时尚的人机界面。



图形用户界面

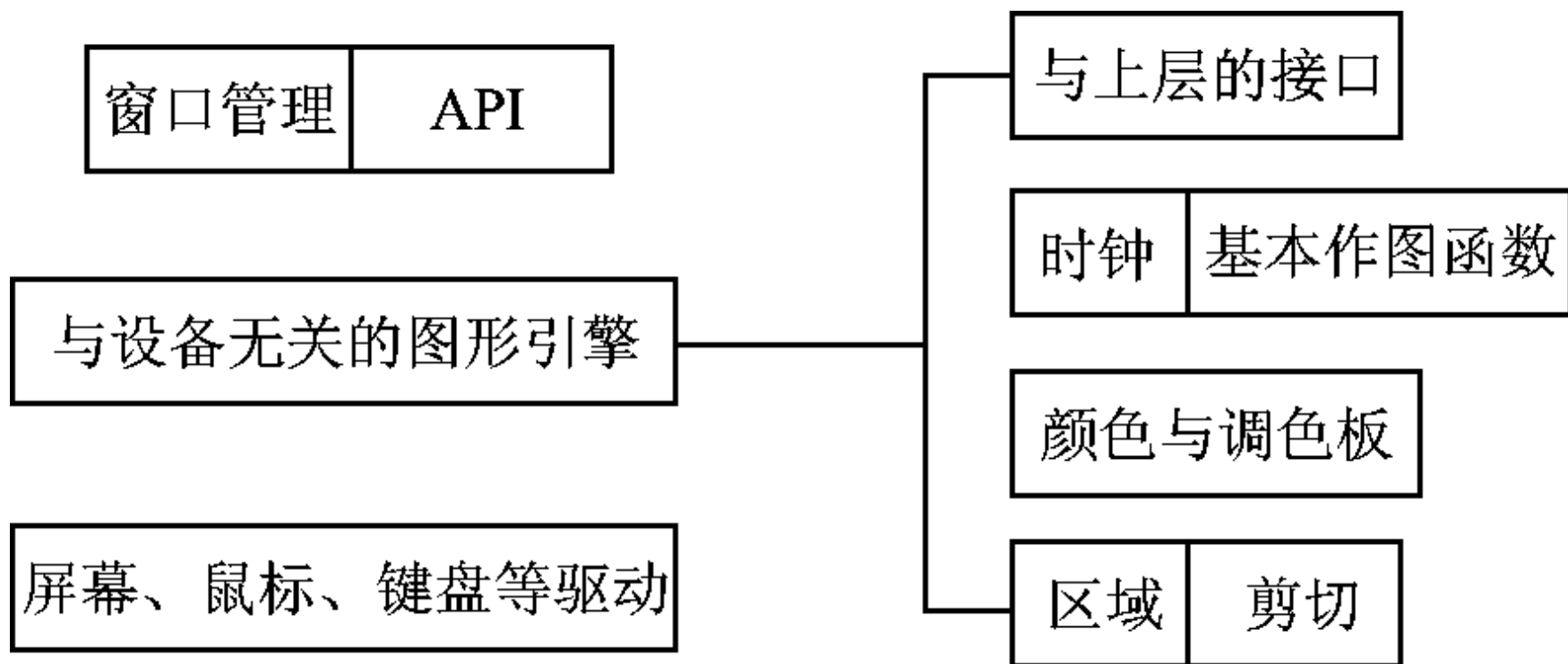
- 图形用户界面-GUI(Graphics User Interface), 它极大地方便了非专业用户的使用和开发人机界面。
- GUI希望是一种WIMP系统。**W**(Windows) 指窗口, 是用户或系统的一个工作区域。一个屏幕上可以有多个窗口。**I**(Icons) 指图标, 是形象化的图形标志, 易于人们隐喻和理解。**M**(Menu) 指菜单, 可供用户选择的功能提示。**P**(Pointing Devices) 指鼠标和触摸等, 便于用户直接对屏幕对象进行操作。
- 用户模型。借鉴不少Desktop桌面办公习惯, 让使用者共享一个直观的界面框架。
- 所见即所得(What You See is What You Get)
- 直接操作。过去的界面不仅需要记忆大量命令, 而且需要指定操作对象的位置。采用GUI后, 用户可直接对屏幕上的对象进行操作, 如拖动、删除、插入以及放大和旋转等。

图形用户界面结构

以典型的MicroWindows为例。

- 充分利用Linux提供的FrameBuffer机制来进行图形显示。Microwindows的可移植性很强。
- Microwindows是基于层式设计的。这样可以根据实际的需要使用不同的层或是对其进行修改。
- 在最底层，屏幕、鼠标、触笔、键盘驱动提供了对实际的显示设备和其他一些用户输入设备的驱动。
- 在中间一层，实现了一个可移植的图形引擎，这个图形引擎实现了画线、区域填充、多边形、剪切和多种颜色模式。
- 在最上层，实现了多种API函数（C/C++），但是这些API很少提供对桌面和窗口外形的支持。

图形用户界面结构



图形用户界面的实现

- 工业计算机需要一个友好方便、稳定可靠的GUI系统，多数采用PC上运行的windows、Linux和嵌入式OS，比如QNX，Vxworks，Intimes。
- 嵌入式设备有严格的资源要求(比如十分有限的存储空间)。同时嵌入式系统经常有一些特殊的要求，而普通的PC和工业计算机上的图形系统是不能满足这些要求的。比如特殊的外观效果，控制提供给用户的函数，提高装载速度，特殊的低层图形或输入设备，以及环境等因素。
- 嵌入式图形界面实现方式多样。
 - MCU+图形控制器或MPU+图形控制器或智能屏。
 - 软件：轻量图形库PEG和 μ C/GUI，有分为内置和外挂。
 - 软件：通用嵌入式OS，如Wince或Linux+Qt或Android。

图形用户界面几个典型工业应用

POS终端



以及很多需要**通信**的应用，
-Web服务器
-多协议桥接器

PLC



以及很多其它**工业/安防/医疗**应用，
- HVAC
- 病患实时监控
- 生化参数测试仪
- 安全监控
- 视频intercom

HMI



以及很多其它**消费/电器**应用，比如
-音频
-安防系统
-扫描仪
-电器

汽车安防



以及很多其它**运输**行业应用，
-汽车后装电子设备
-GPS/船舶导航

图形用户界面的重要性

- **让 OEM 创建自己设备的UI 品牌**
 - UI 可以匹配产品的价值；酷、容易使用、创新和可靠等。
 - 需要很快的设计出一个很酷很复杂的UI。
 - UI的差异化-与实现方式有关。
 - UI 最好是可复用和跨设备和OS平台。
- **UI的可用性是关键**
 - 需要很快速的设计原型，测试和修改UI。
 - 需要一个不受工程资源约束的UI。
- **帮助获得利润**
 - UI 可以用在不同的设备上。
 - 使产品更加吸引客户。
 - 帮助产品脱颖而出。

OpenGL/ES和3D UI



- OpenGL是3D图形接口的工业标准，是“原始”3D图形、由GPU加速，在不同GPU上提供标准化的底层接口，创建自定义顶点和片段着色器以产生特效，非常好的性能 其他所有框架最后都会调用OpenGL进行渲染。
- OpenGL ES是专为嵌入式和移动设备2D/3D轻量图形库，它是基于OpenGL API设计，支持低功耗设计。
- OpenGL ES 图形加速已经在许多GPU上实现了包括 Imagination Power VR, ARM Mali, Vivante GCxxx, Qualcomm Adreno。
- Android、QNX 6.5等都支持OpenGL/ES。
- 3DUI 已经在智能电视和家庭娱乐设备等嵌入式电子产品中使用。

今天的Android UI

- 许多的 OEMs/ODMs 正在开发 Android 设备。
- 用户的体验很重要
 - 赢得客户和获得利润的手段。
- UI设计是非常必要的，但难以达到根治的变化。
- HTC, Sony Ericsson 和 Motorola 已经在这个领域投资许多年了。
 - 他们认识在Android 平台，差异化很重要。
- 手机的例子: 富有创新的用户正在开发3D UIs



Sony Ericsson XPERI
的Rachel'

Motorola's MotoBlur UI. 专注社交网络

HTC Hero 是私有
"Sense" UI

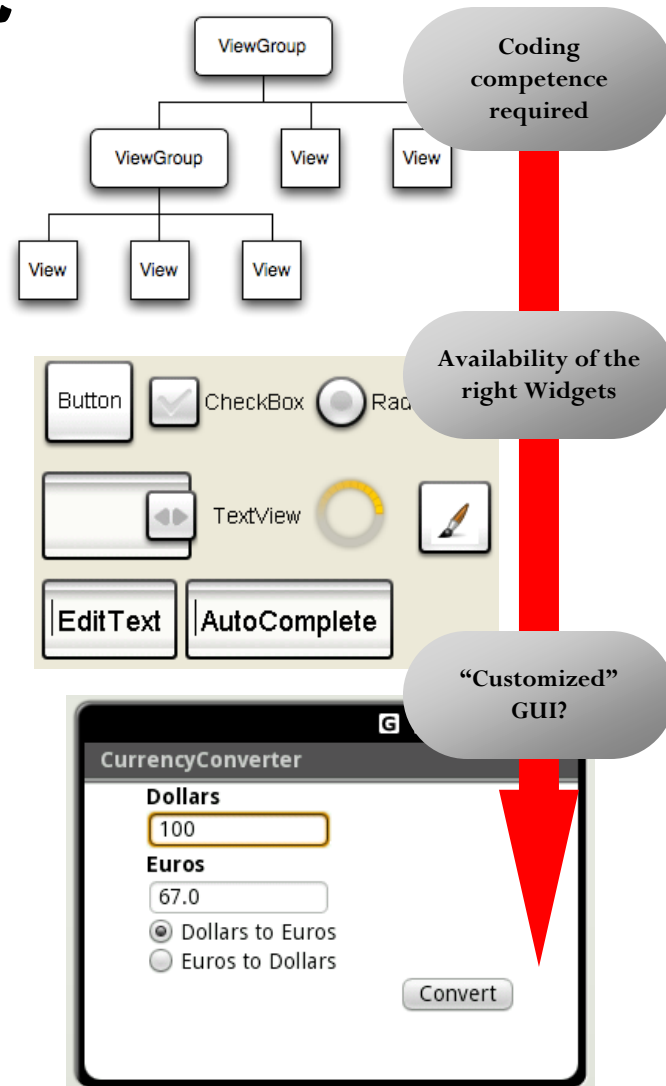
Android UI 用户化的困境

■ 很难产生光滑的动画用户界面

- 在Android的唯一的选项是 hard code
 - 需要Open GL/ES APIs**细节知识**。
 - **花费很长的时间**。
 - 导致UI的配置很少，因此为未来产品需要昂贵的升级。

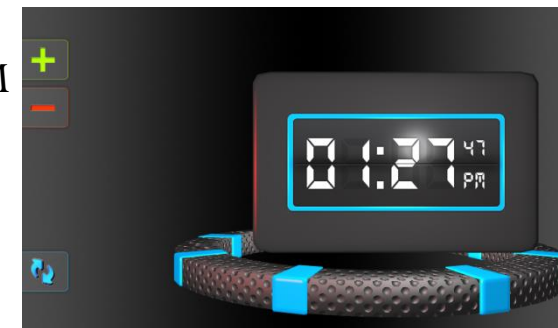
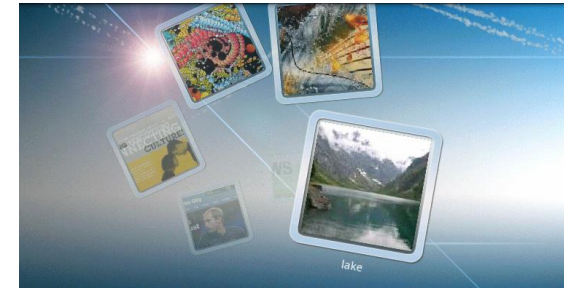
■ 耗时和昂贵的创建非动画的UI

- **需要编程**
 - **耗时间, 很困难快速改变视觉**
 - 有限的范围内的显著变化



支持Android的Inflexion UI

- 在Android上加入Inflexion Eclipse
 - 用鼠标就可以轻松的开发基于Android Inflexion应用。
 - 自动的生成基于Inflexion + Android APIs Activity 和 framework classes 。
 - 自动生成Java 接口, 模块集成代码和 make file文件。
- 与Android深度集成
 - 基于Java based 模块接口, 提供模块接口充分利用底层Android框架。
 - 自动构建成C/JNI 代码。
- 从根本区分设备
 - 例子: **Home Screen Replacement** –提供丰富, 平滑的动画UIs , 支持widgets, applications 和shortcuts
 - 例子: **Contact List / Phonebook Application** – 提供给OEM无限的定制机会。

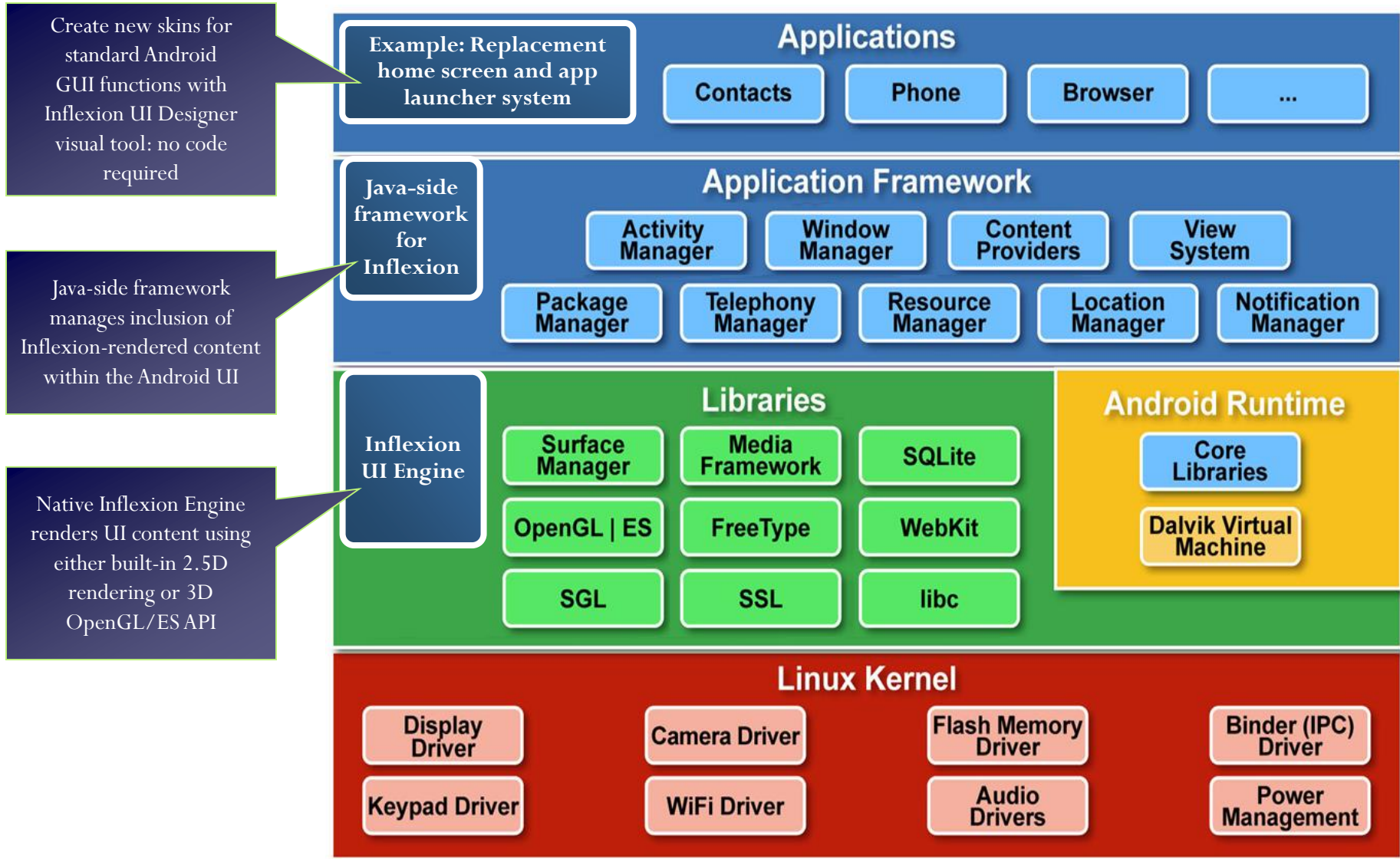


Inflexion的特点

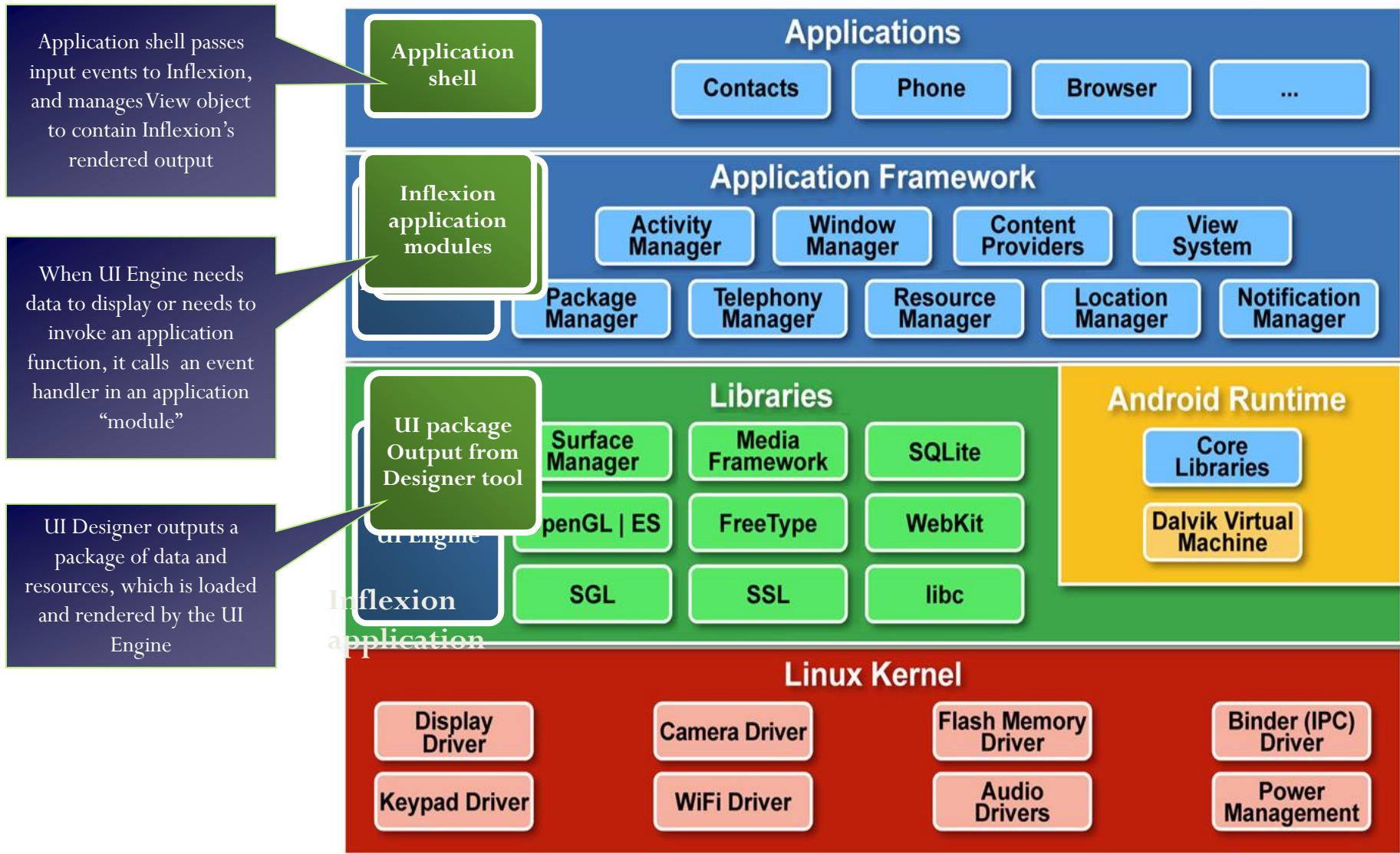
- **Drag 和drop UI 架构**
 - 使得创建动画设计更快和方便。
- **3D 和动画**
 - 创建动画和3D Uis，不需要理解 OpenGL/ES。
- **在 PC上预览和进行交互UI设计**
 - 改善工作流程，避免设计，编码，测试周期过长。
- **把UI与应用代码区分开。**
 - UI 可以从应用代码入手独立开发。
- **可伸缩和可移植**
 - 即可以使用在低端和没有图形加速芯片上，也可使用在高端加速GPU上。
- **开发费用和版税，不适合已有大量UI设计团队，要求更加深入和灵活。**

允许应用跨产品线范围的复用

如何与 Android 协调工作



Inflexion 应用如何工作



未来UI软件发展的趋势



Thank You !

