



移动应用开发 案例研究

Wei Dong



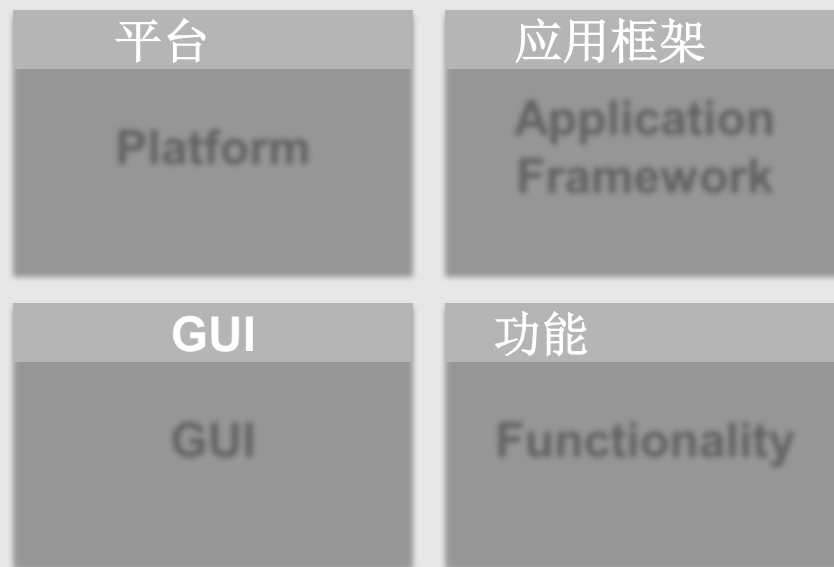
内容

- **Tizen** 应用开发
- **Tizen** 应用开发实践
 - 性能优化
 - 内存使用
 - 数据库使用

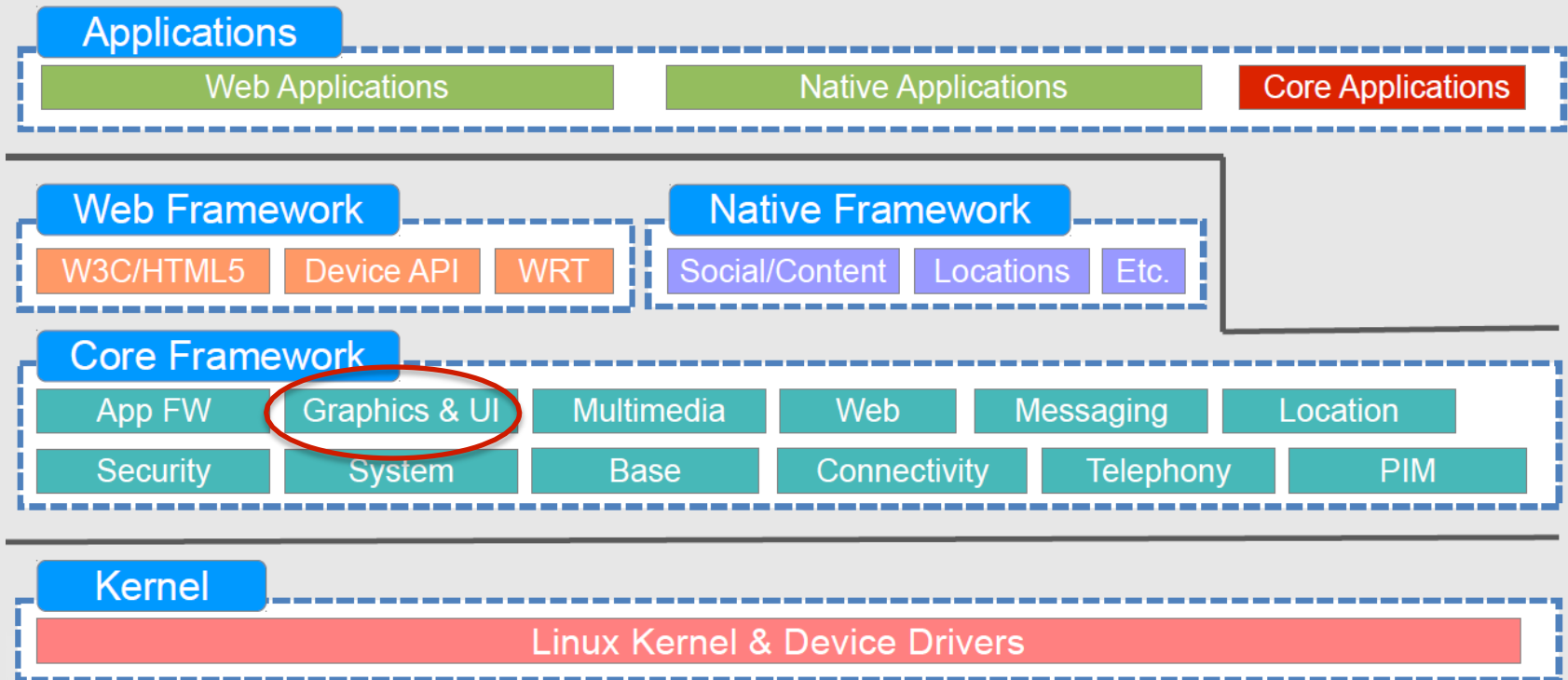


Tizen 应用开发

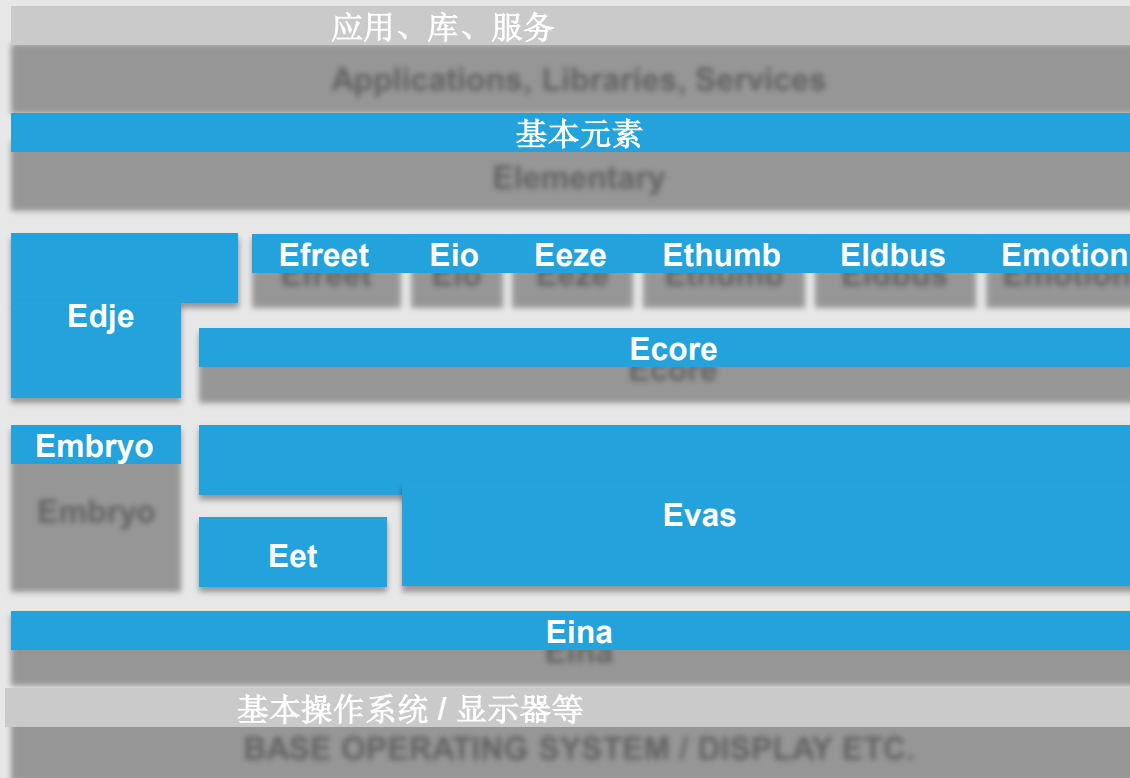
应用开发 - 从这里开始



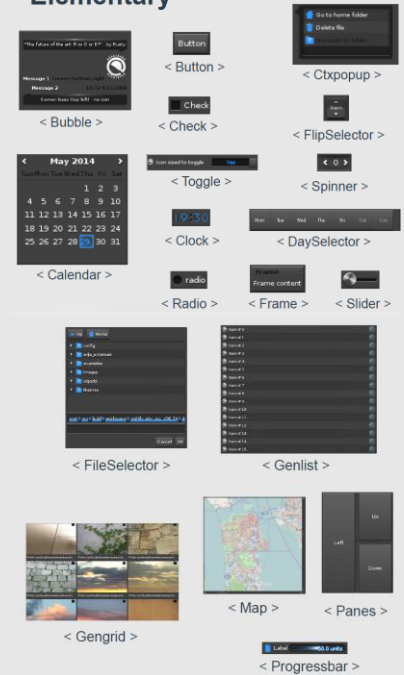
Tizen 架构



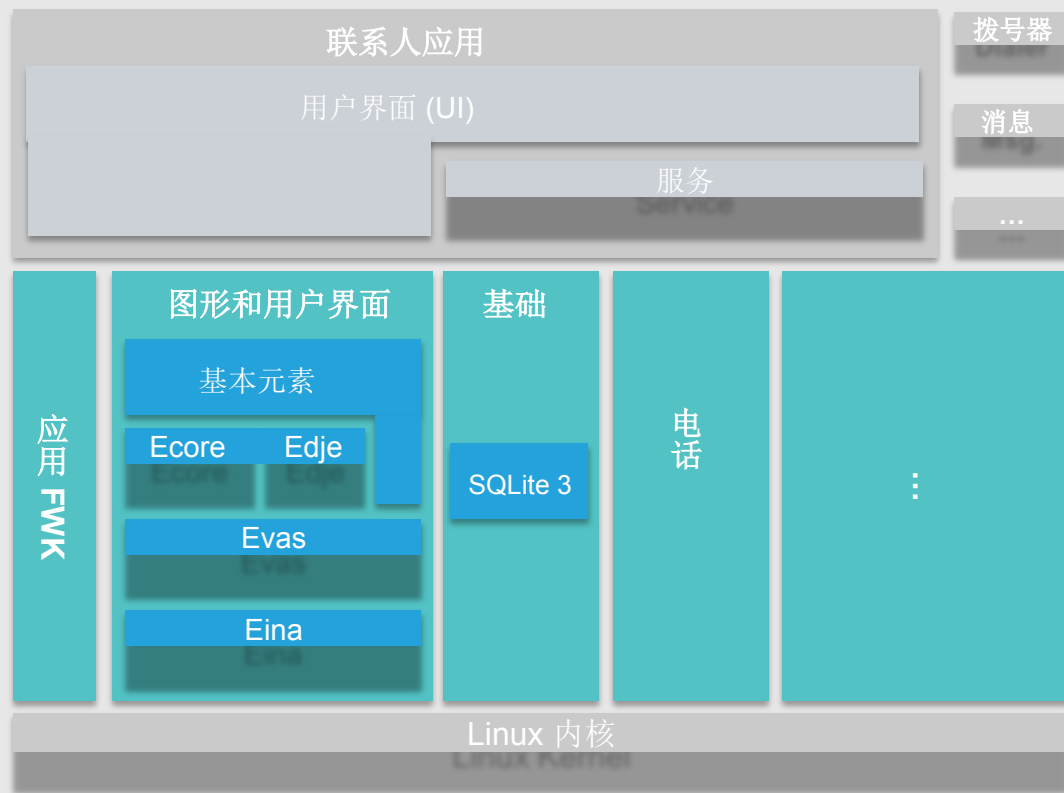
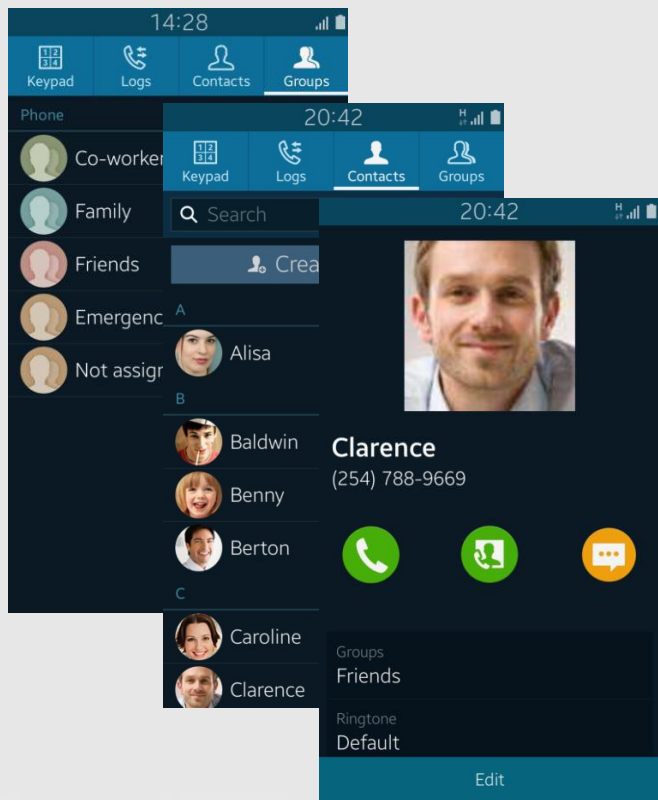
EFL 架构



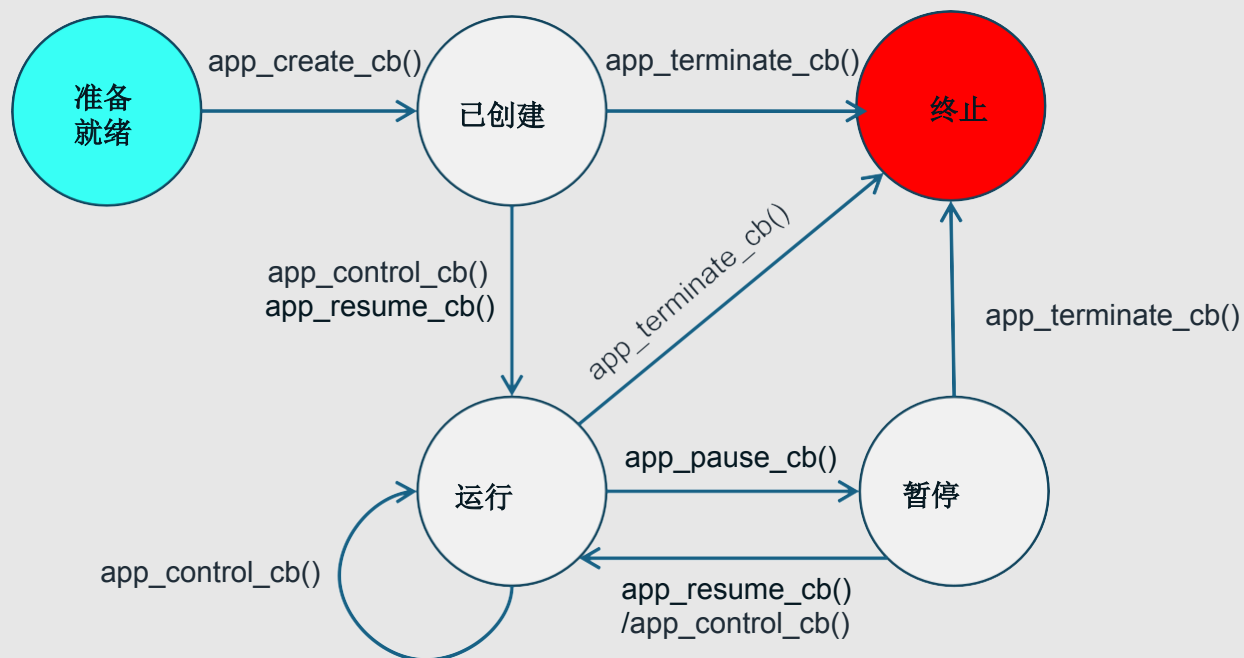
Elementary



案例研究：联系人应用



应用生命周期

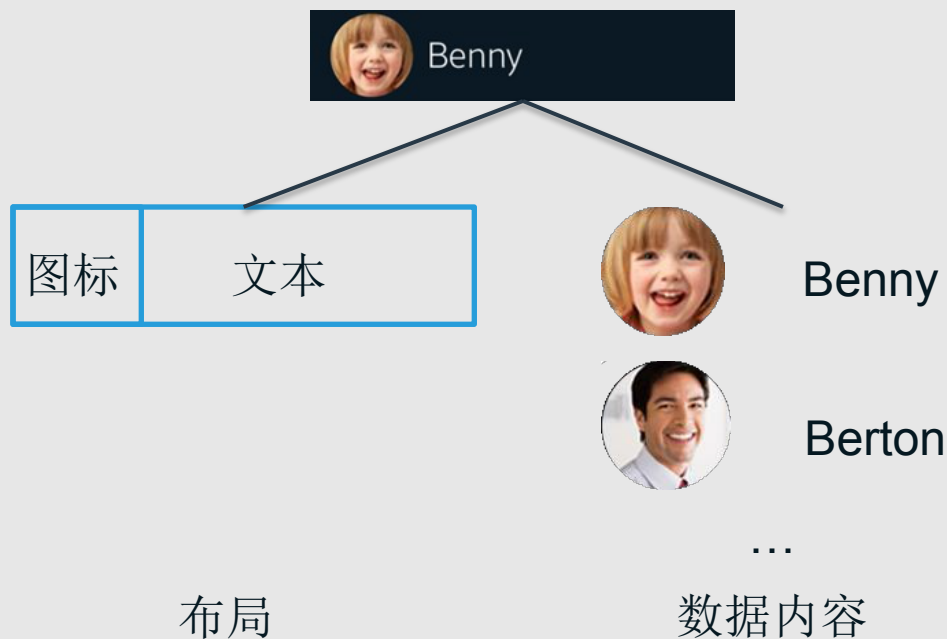
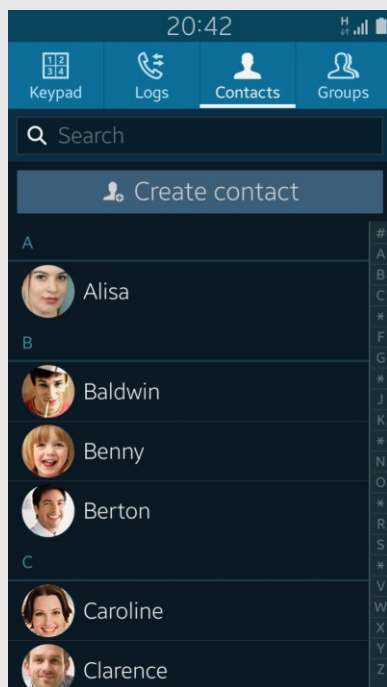


main() 示例代码

```
int main(int argc, char *argv[])
{
    struct appdata ad; app_event_callback_s event_callbacks;
    event_callbacks.create = ct_create;
    event_callbacks.terminate = ct_terminate;
    event_callbacks.pause = ct_pause;
    event_callbacks.resume = ct_resume;
    event_callbacks.app_control= ct_app_control;
    ...
    return app_efl_main(&argc, &argv, &event_callbacks, &ad);
}
```

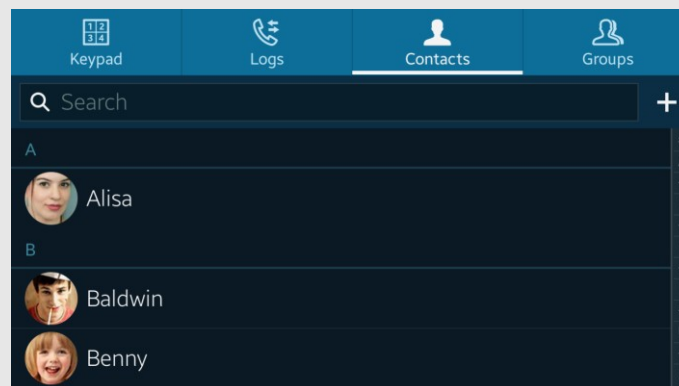
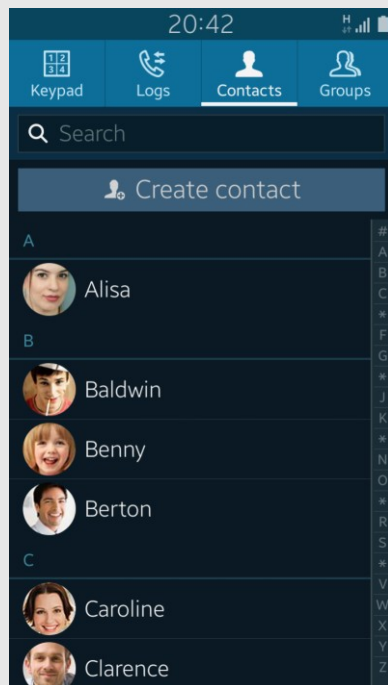
用户界面 (UI) 分析

- 布局
- **Widget** 选择



可扩展性

- 相对位置



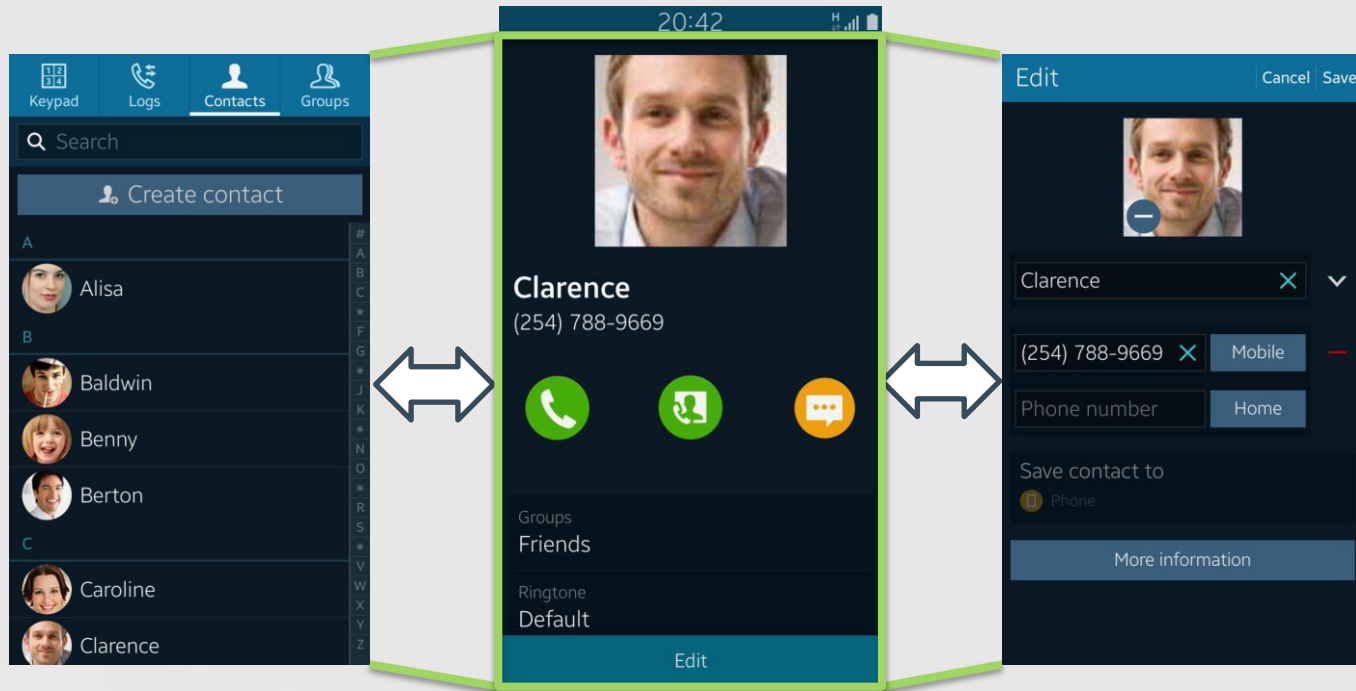
示例代码

```
group {  
  name:"item"; parts {  
    part {  
      name:"icon";  
      type:SWALLOW;  
      description {  
        state:"default" 0.0;  
        min:72 0;  
        align:0.0 0.5;  
        fixed:1 0;  
        rel1 { relative:0.0 0.0 ;}  
        rel2 { relative:0.0 1.0 ;}  
      }  
    }  
  }  
}
```

```
part {  
  name:"text"; type:TEXT; description {  
    state:"default" 0.0;  
    rel1 { relative:1.0 0.0 ; to:"icon";}  
    rel2 { relative:1.0 1.0 ;}  
  }  
  ...  
}
```

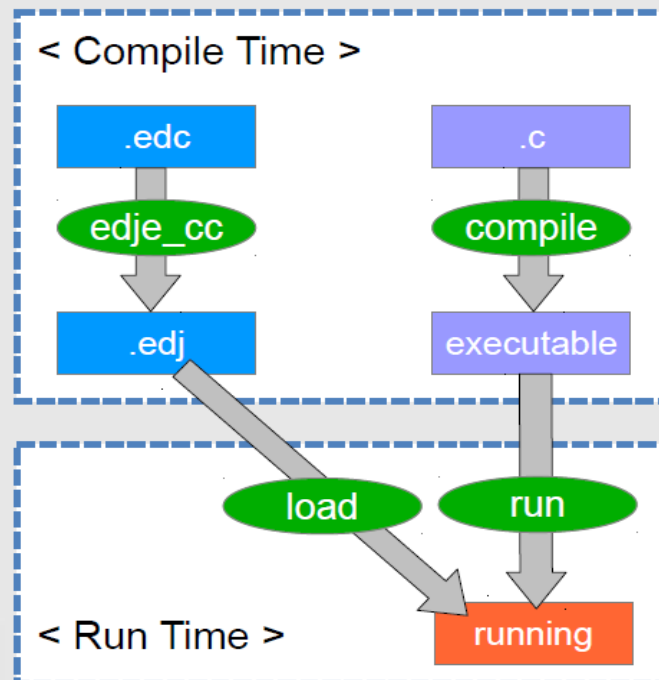
应用视图

- 借助导航框架进行视图控制



构建应用

- 布局与逻辑相分离
 - 图形部分：GUI 二进制布局 (edj)
 - 功能：可执行的二进制逻辑 (C)





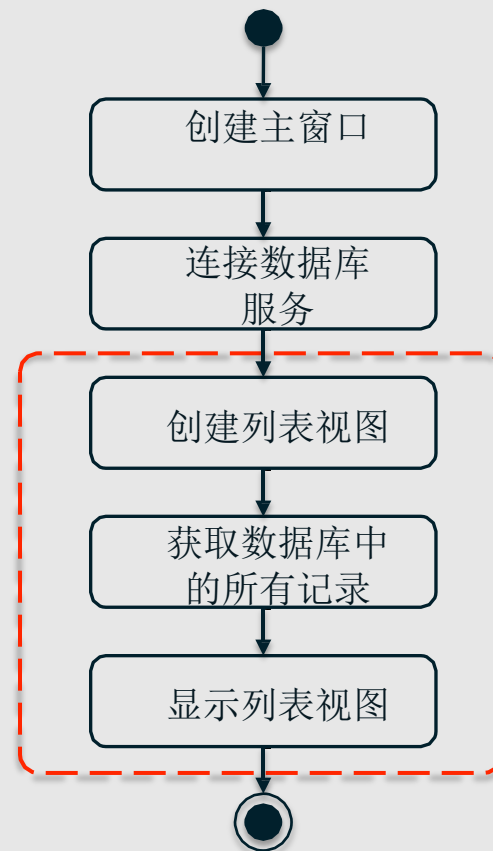
性能优化实践



启动性能分析

- 同步过程
 - 海量数据分析性能较差

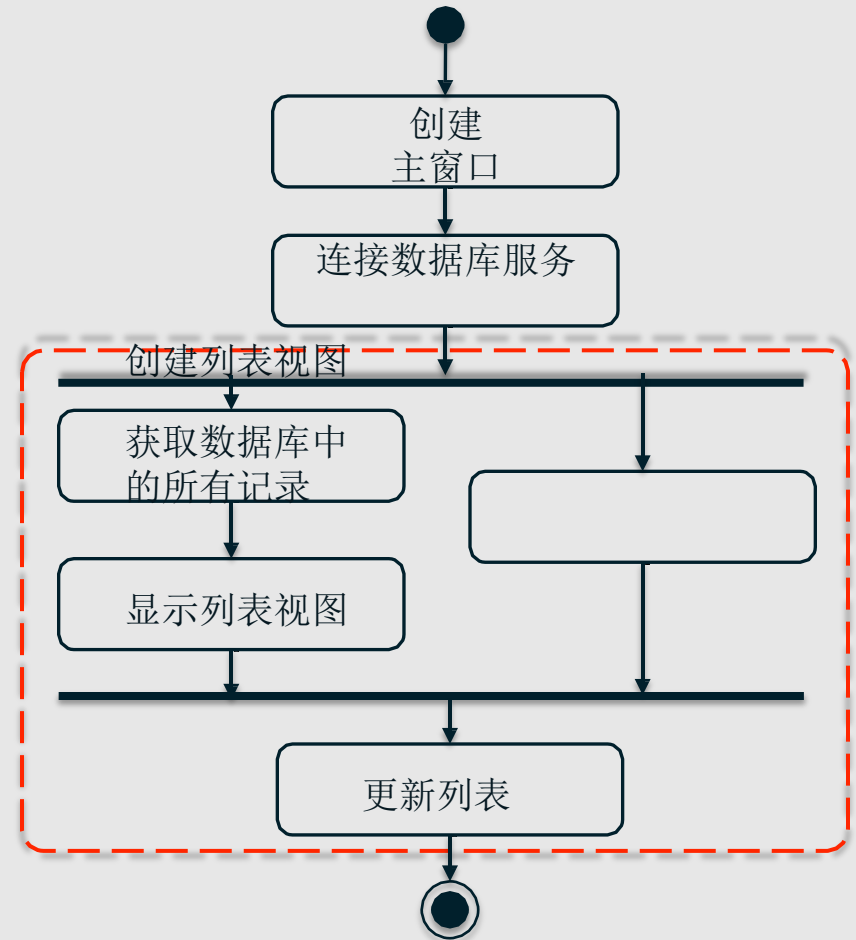
应用	条件	启动时间
联系人	10000 条记录	2.27 秒



并行处理

- 多线程处理
 - 异步检索数据

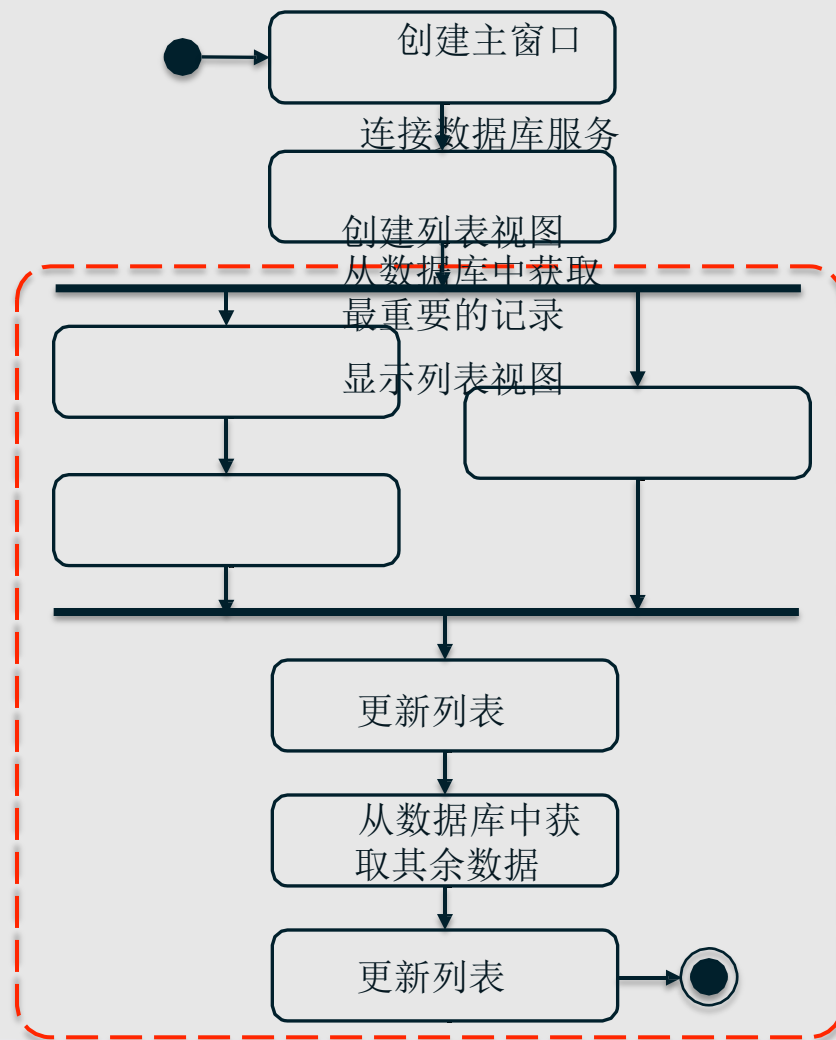
应用	条件	启动时间
联系人	10000 条记录	1.53 秒



进一步优化

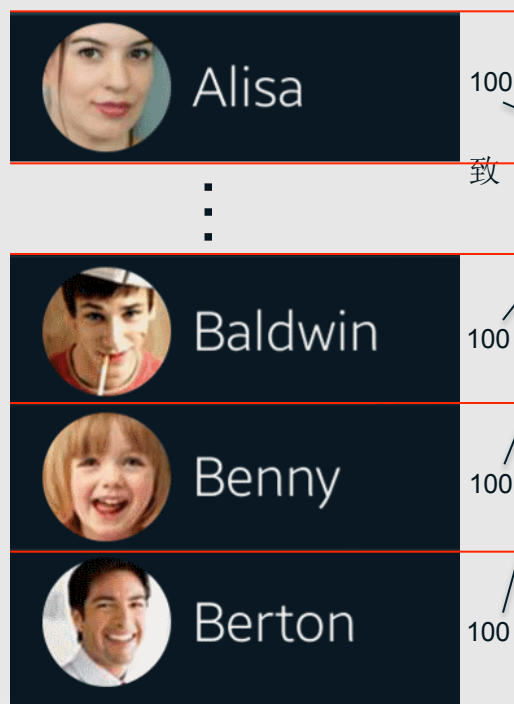
- 分割数据加载
 - 检索必要的显示数据
 - 在闲置状态下加载其余数据

应用	条件	启动时间
联系人	10000 条记录	1.04 秒



列表项加载

- 借助 `elm_genlist_homogeneous_set()` 创建列表项



列表项应保持高度一致

```
...  
Evas_Object *ct_create_genlist(...)  
{  
...  
genlist = elm_genlist_add (parent);  
elm_genlist_homogeneous_set(genlist, EINA_TRUE);  
... elm_genlist_item_append(...);  
...  
}
```



内存使用实践

Tizen 中的内存工具

- **memps**

- memps [-a] | [-v] | [-s] <pid> | [-f] <输出文件的完整路径>

```
sh-3.2# memps -s 856
S(CODE)  S(DATA)  P(CODE)  P(DATA)  SHARED(CODE+DATA)  TOTAL
-----
14196    2720     3312     45192    2384              50888
```

	描述
S(CODE)	共享的清洁内存，其包括复制的内存
S(DATA)	共享的脏内存，其包括复制的内存
P(CODE)	私有清洁内存 (clean memory)
P(DATA)	私有脏内存 (dirty memory)
PEAK	S(CODE) + S(DATA) + P(CODE) + P(DATA) 的内存使用峰值

	描述
PSS	比例设置大小
3D	GPU 驱动程序分配的内存
UMP	分配给显卡的内存
GEM	通过 GEM 分配的内存 (ALLOC: 不包括复制的内存)
VIDEO	分配给帧缓冲区的内存
TOTAL	PSS + 3D + GEM(ALLOC) + VIDEO

Tizen 中的内存工具

- **Valgrind**

- 面向内存调试、内存泄漏检测和分析

空指针解引用

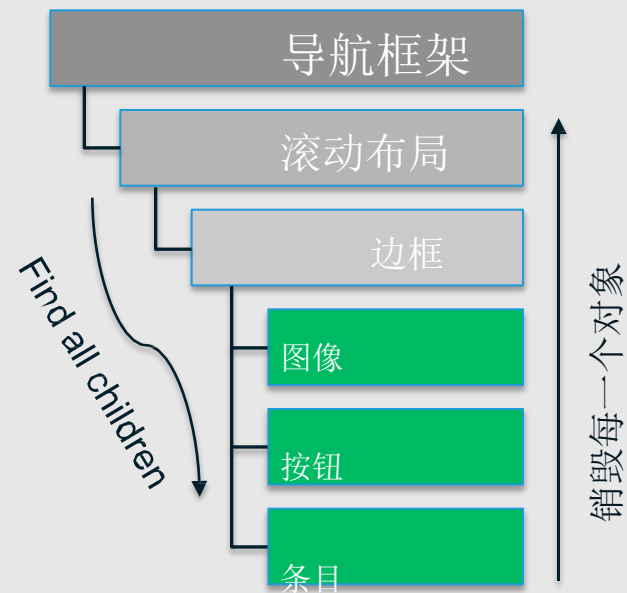
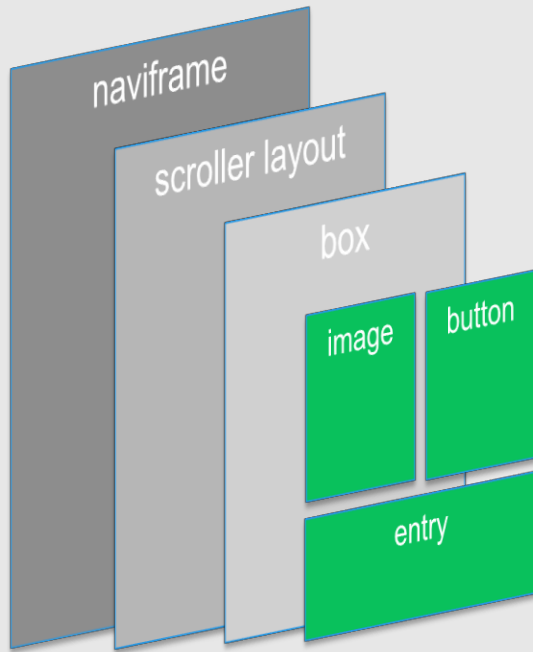
```
==2741== Invalid write of size 4
==2741==    at 0x57700B0: ct_input_push_navi_content (ct-input-view.c:7694)
==2741==    by 0x5752A29: ct_input_launch_preloaded_view (ct-detail-main.c:833)
==2741==    by 0x4953579: ct_list_create_contact_btn_clicked_cb (ct-list-contact-view.c:539)
==2741==    by 0x4D2E83F: evas_object_smart_callback_call (in /usr/lib/libevas.so.1.7.99)
==2741==    Address 0x0 is not stack'd, malloc'd or (recently) free'd
```

内存泄漏

```
==11097== 1,024 bytes in 1 blocks are definitely lost in loss record 144 of 196
==11097==    at 0x4834FBC: malloc (vg_replace_malloc.c:291)
```

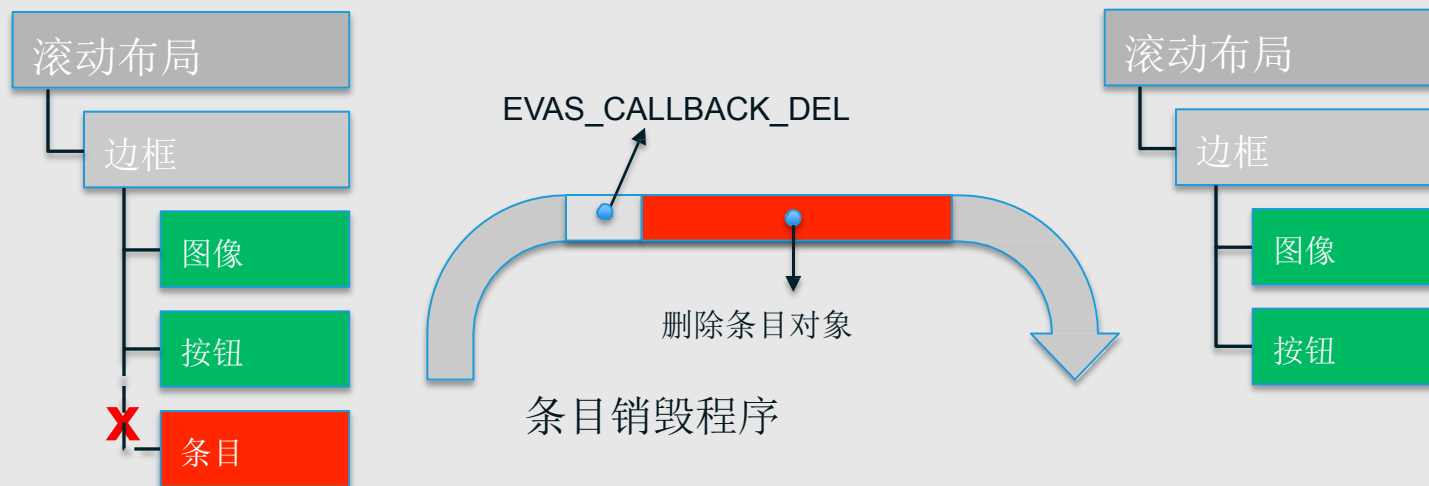
Widget 对象发布

- widget 对象的生命周期



用户数据发布

- 对象销毁中的事件
 - `EVAS_CALLBACK_DEL`

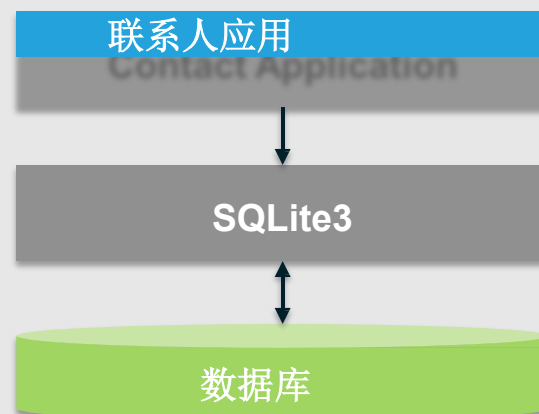




数据库使用实践

Tizen 中的数据库引擎

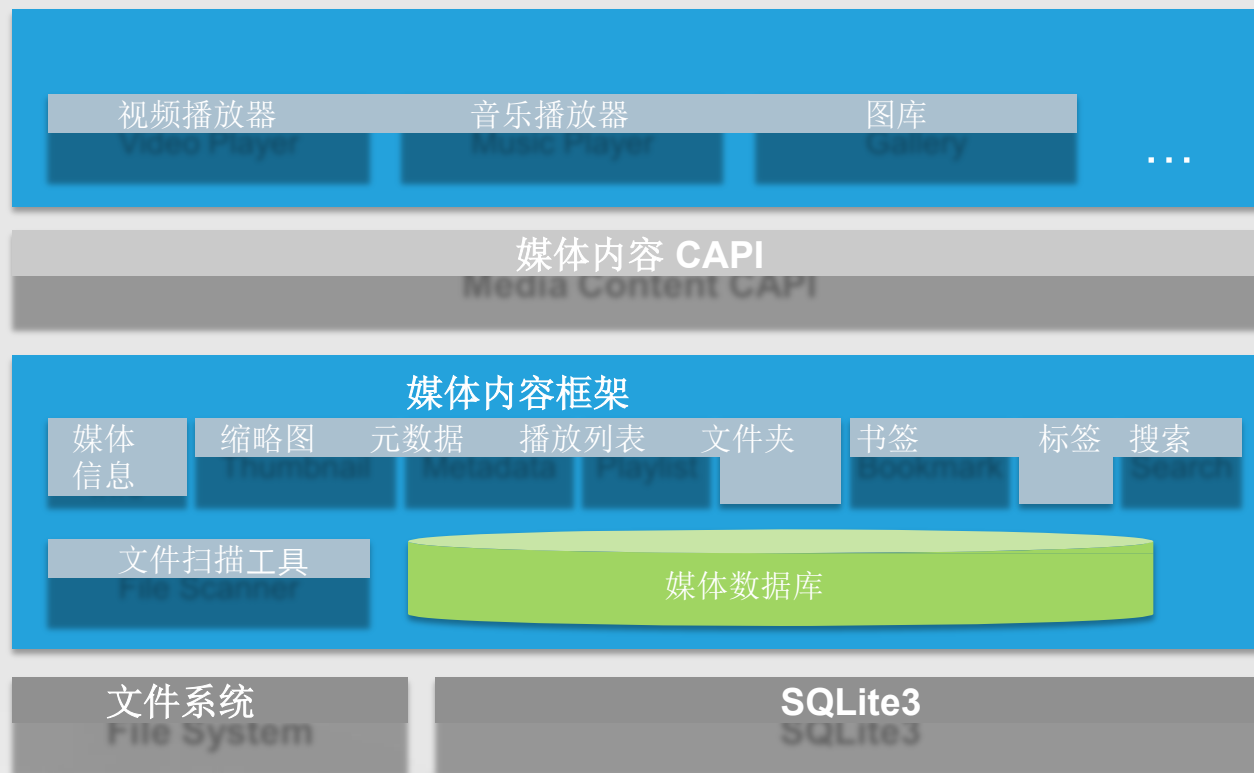
- **SQLite 3**
 - 倍受欢迎的嵌入式数据库
 - Tizen 平台中没有打包工具（wrapper）
- **SQLite3 实践**
 - 并发控制
 - 在应用端使用互斥锁
 - 执行 SQL 语句
 - `sqlite3_exec()`
 - `sqlite3_prepare_v2()` & `sqlite3_step()`



媒体内容框架

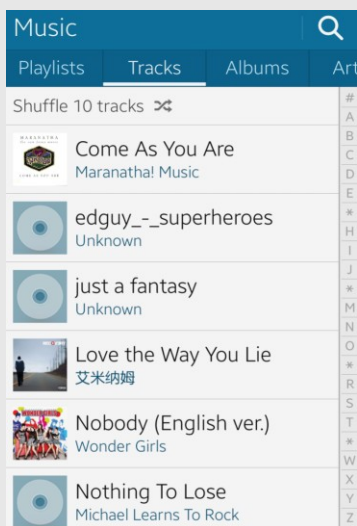
- **CRUD**（创建、读取、更新和删除）数据库的媒体信息
- 管理内容：图像/视频/音频文件

应用



案例研究：带有媒体内容的音乐播放器

获取媒体信息



```
...
/* 初始化 */
media_content_connect(...);
/* 创建过滤器 */
filter = media_filter_create(...);
/* 获取音频文件数 */
media_info_get_media_count_from_db(..., &count);
...
/* 设置过滤器属性 (SQL 准备) */
media_filter_set_order(...);
media_filter_set_condition(...);
media_filter_set_offset(...);
/* 查询媒体内容中的音频文件列表 */
media_info_foreach_media_from_db(..., filter);
...
/* 监控音频文件更新 */
media_content_set_db_updated_cb(...);
...
/* 初始化 */
media_content_disconnect(...);
...
```

获取元数据

```
...
/* 初始化 */ media_content_connect(...);
media_info_get_media_from_db(...);
media_info_get_audio(...);

audio_meta_get_sample_rate(...);
audio_meta_get_duration(...);
...
/* 初始化 */
media_content_disconnect(...);
...
```



Over the Horizon
Samsung

Artist Samsung
Title Over the Horizon
Album Brand Music
Track length 02:56
Genre Symphony Rock
Author Unknown
Copyright Unknown
Track number Unknown
Format 128Kbps 44.1KHz 2ch
Location Device memory/Sounds/Over the horizon.mp3

问答



TIZENTM

DEVELOPER SUMMIT 2014



SHANGHAI



TIZEN开发者峰会（上海）