

我在大厂做研发

我在百度做阅读器

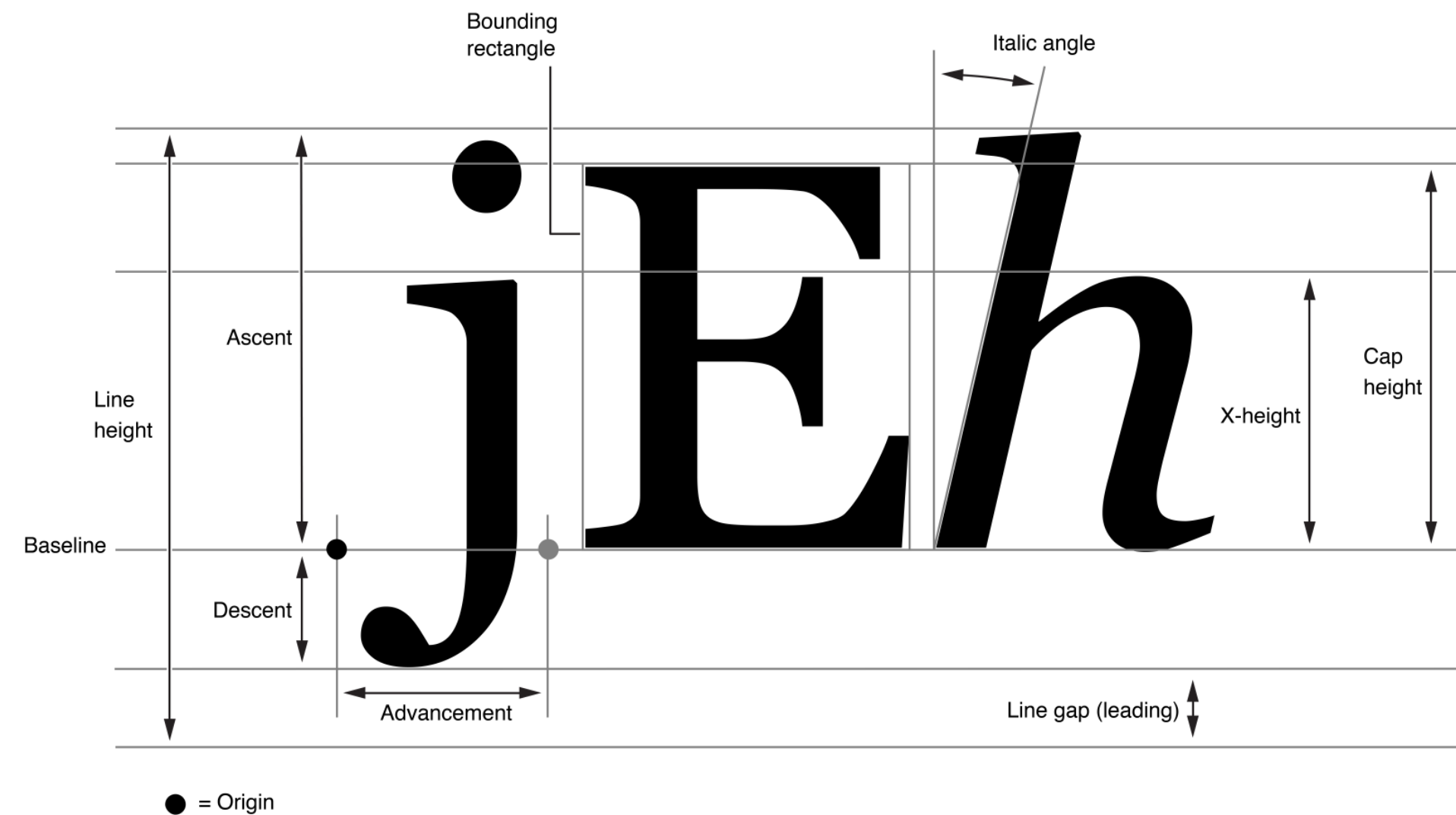
Kyo @Baidu

T Chat 什么是排版?



➤ 系统是如何帮我们把文字排列的?

场景 1 通过查找 app 找到它们
通过查找 app 找到它们



场景 2 排版：在固定页面内，将内容以适合的方式呈现。
这就是排版的定义。

排版：在固定页面内，将内容以适合的方式呈现。
这就是排版的定义。

广义排版引擎： 排版+页面渲染的引擎，它解析输入的内容（HTML、XML等等）及其属性，根据页面布局，并运用规则，把内容排到合适的位置进行渲染、显示。

狭义排版引擎： 仅指广义排版引擎中排版部分。

流式

块式

天龙八部（世纪新修版）

“金庸作品集”新序

小说是写给人看的。小说的内容是人。

小说写一个人、几个人、一群人、或成千成万人的性格和感情。他们的性格和感情从横面的环境中反映出来，从纵面的遭遇中反映出来，从人与人之间的交往与关系中反映出来。长篇小说中似乎只有《鲁滨逊漂流记》，才只写一个人，写他与自然之间的关系，但写到后来，终于也出现了一个仆人“星期五”。只写一个人的短篇小说多些，尤其是近代与现代的新小说，写一个人在与环境的接触中表现他外在的世界、内心的世界，尤其是内心世界。有些小说写动物、神仙、鬼怪、妖魔，但也把他们当作人来写。



华主席会见萨莫拉总统

华主席对萨莫拉总统说：我们坚决支持你们前线国家，大力援助南部非洲人民的解放斗争。

新华社北京五月二十八日电 中共中央主席、国务院总理华国锋今天下午会见莫桑比克解放阵线主席、莫桑比克人民共和国总统萨莫拉·莫伊塞斯·马谢尔时说：“你们来自南部非洲斗争的前线，我们坚决支持你们前线国家，大力援助南部非洲人民的解放斗争。”



华主席说：“萨莫拉总统是我们的老朋友了。我们两国的关系是很亲密的。在你们为争取民族独立进行武装斗争的时候，我们就和你们站在一起。”华主席表示相信，获得了民族独立的莫桑比克人民，在萨莫拉总统领导下，一定能够克服遇到的困难，取得更大的胜利。萨莫拉总统说：“我们的关系是在莫桑比克人民斗争艰苦的岁月里建立起来的。我们的关系是牢固的。”

谈话时在座的莫桑比克贵宾还有若阿金·希萨诺外长、马里奥·马顺戈部长、若泽·路易斯·卡巴索部长、塞尔吉奥·维埃拉部长。

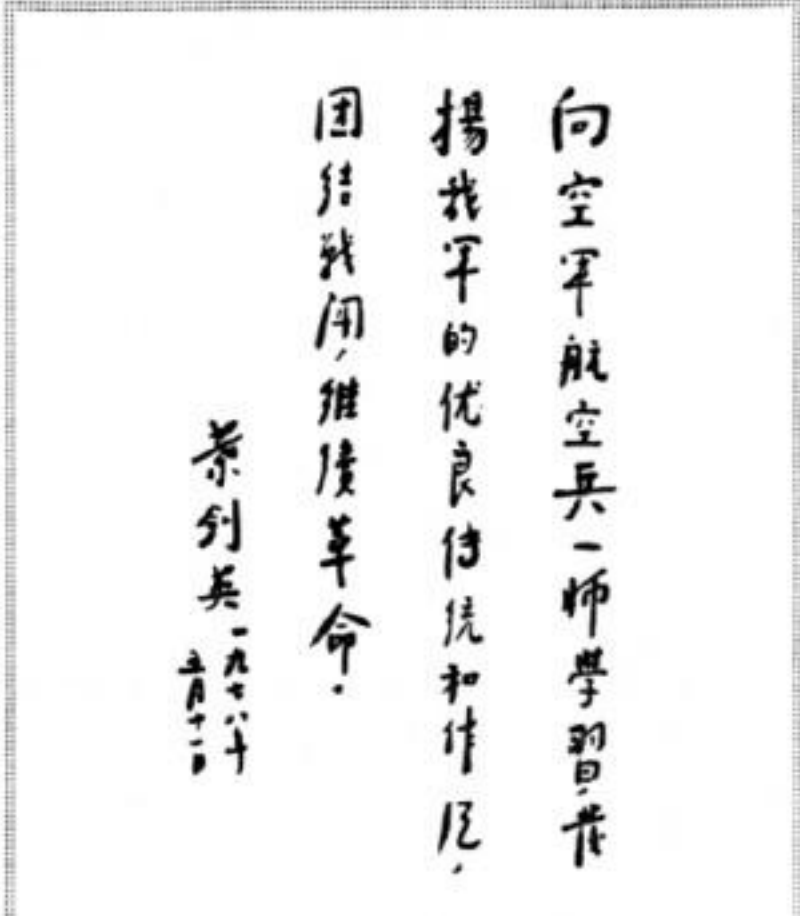
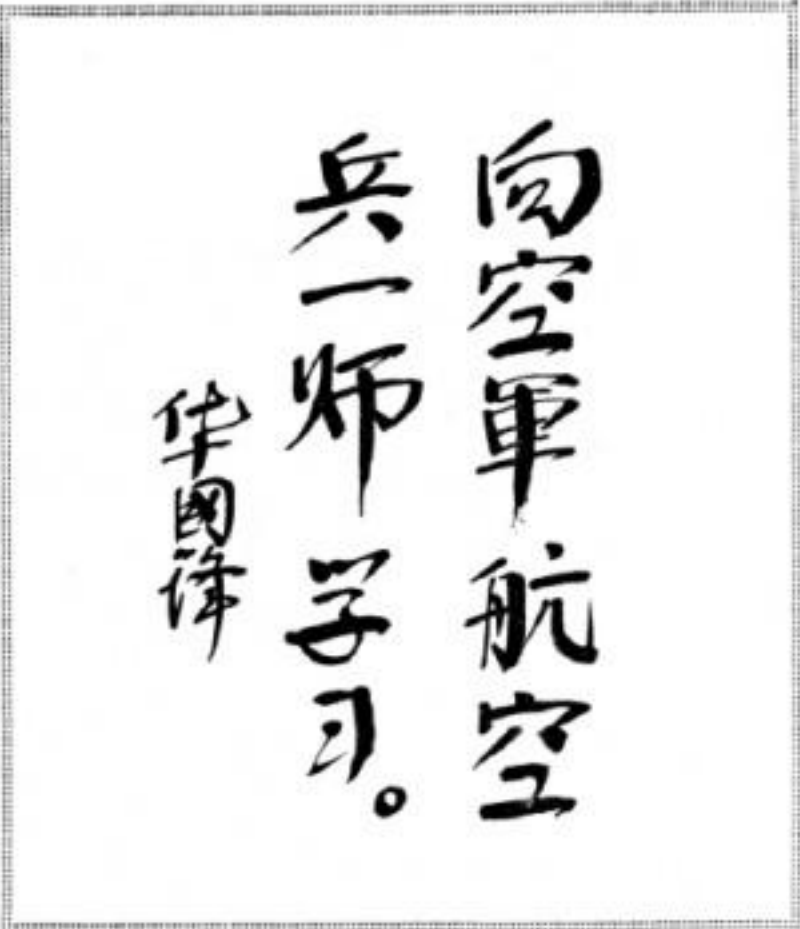
李先念副总理、何英副外长、张才千副总参谋长、程飞副部长、中国驻莫桑比克大使杨守正，参加了谈话。

华主席同萨莫拉总统握手。

新华社记者摄

华主席叶副主席为学习空军航空兵一师题词

还分别为空军题词 空军领导机关和航一师分别集会决心认真贯彻题词精神



英明领袖华主席为学习空军航空兵一师题词手迹。

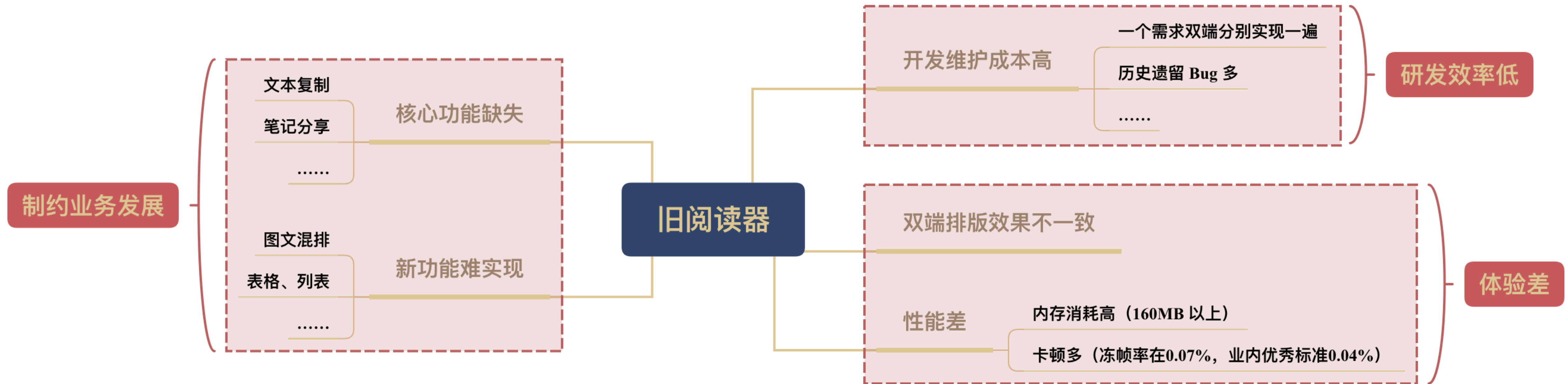
敬爱的叶副主席为学习空军航空兵一师题词手迹。

新华社发

T Chat 小说新 C++ 排版内核诞生的背景

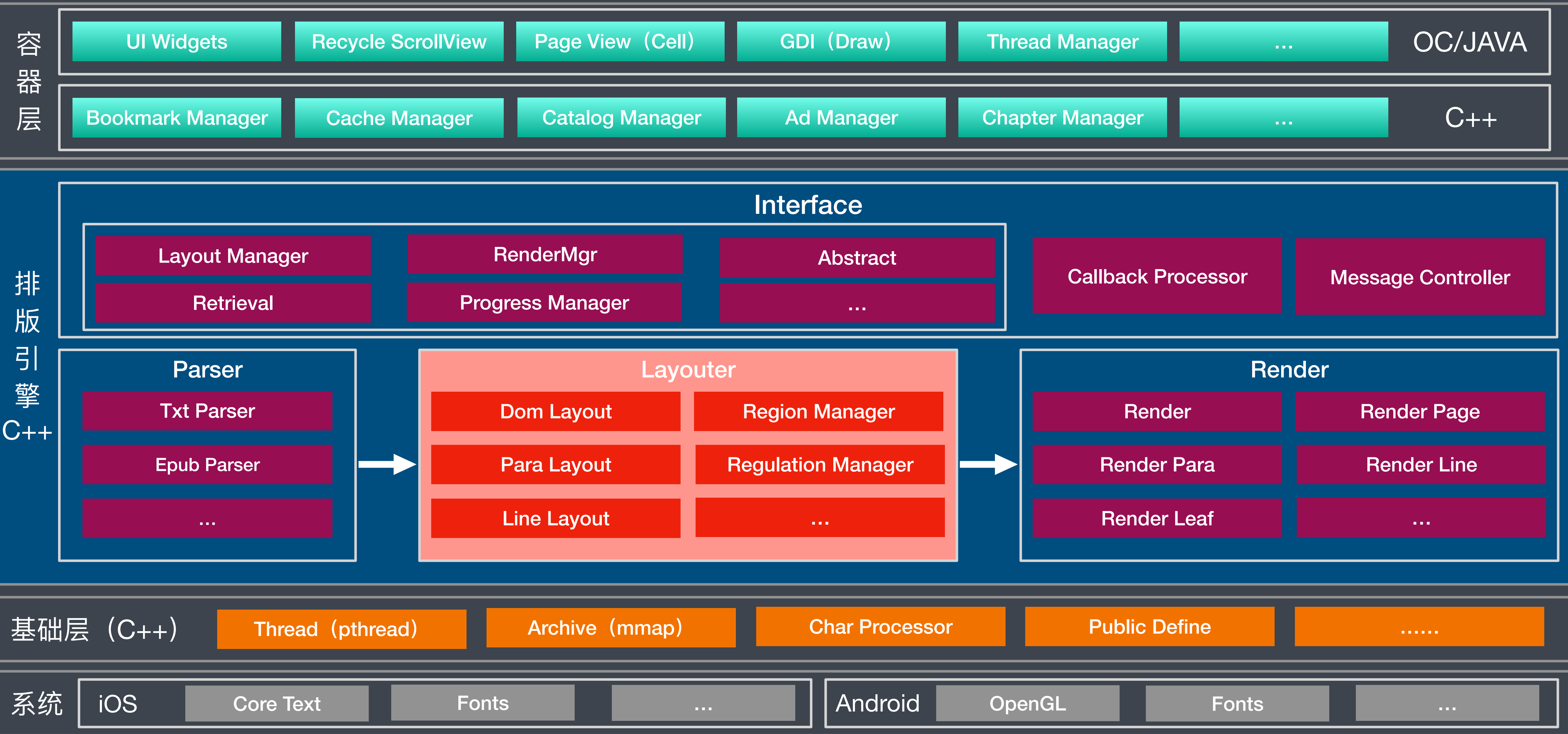


背景：iOS、Android 两套阅读器，排版能力差，扩展性低

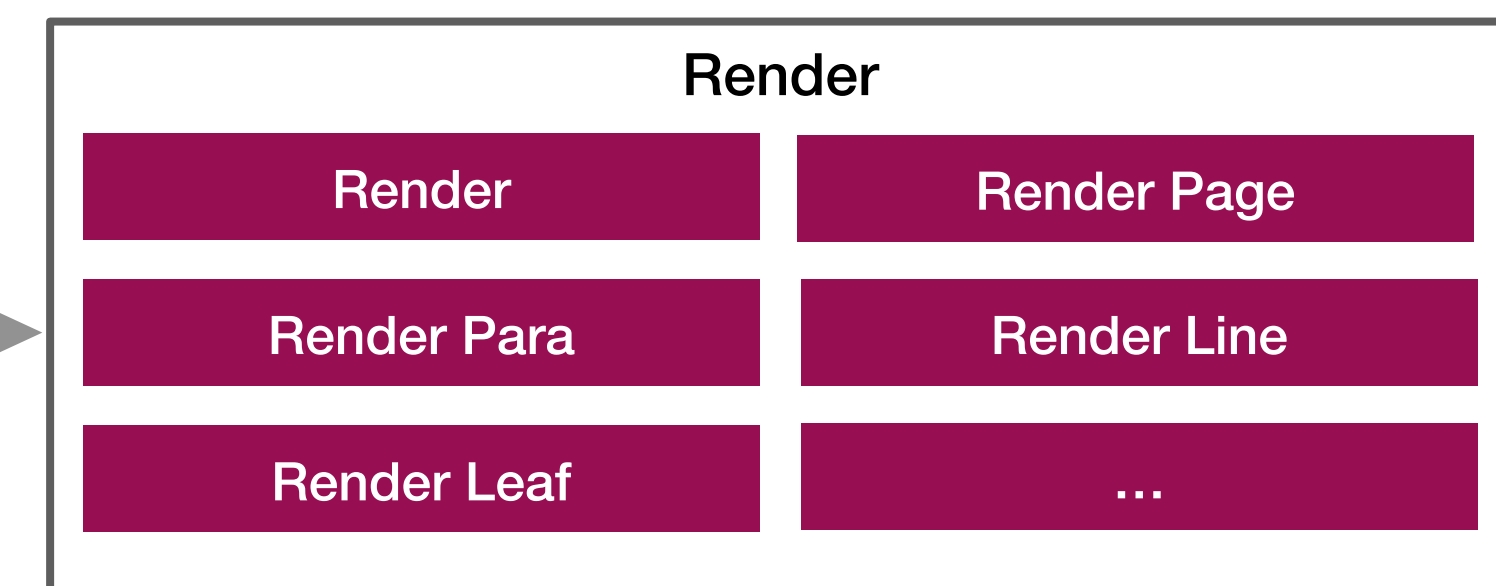
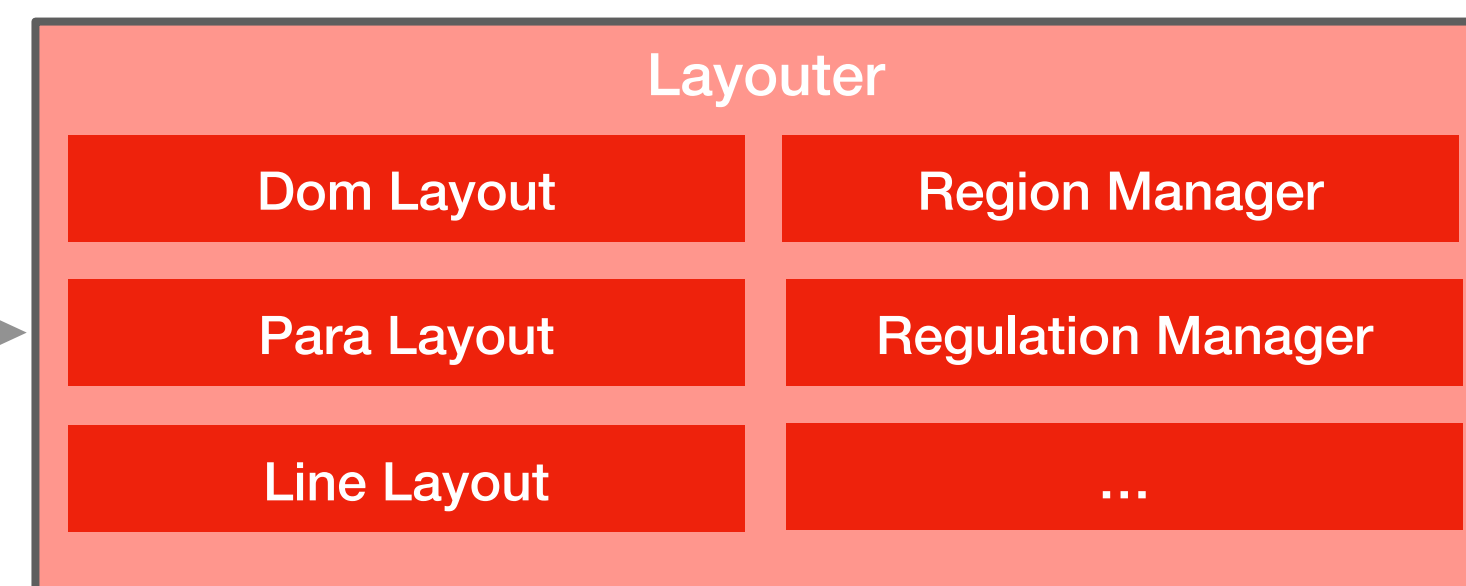
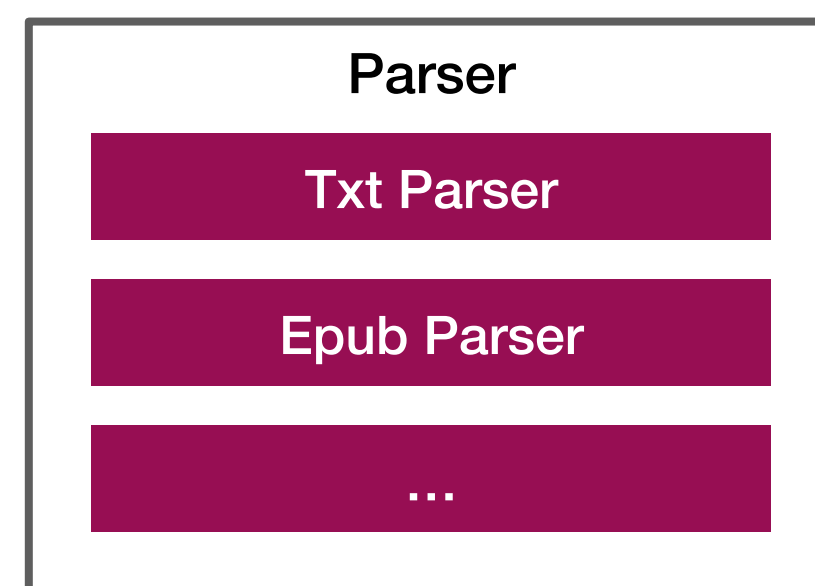
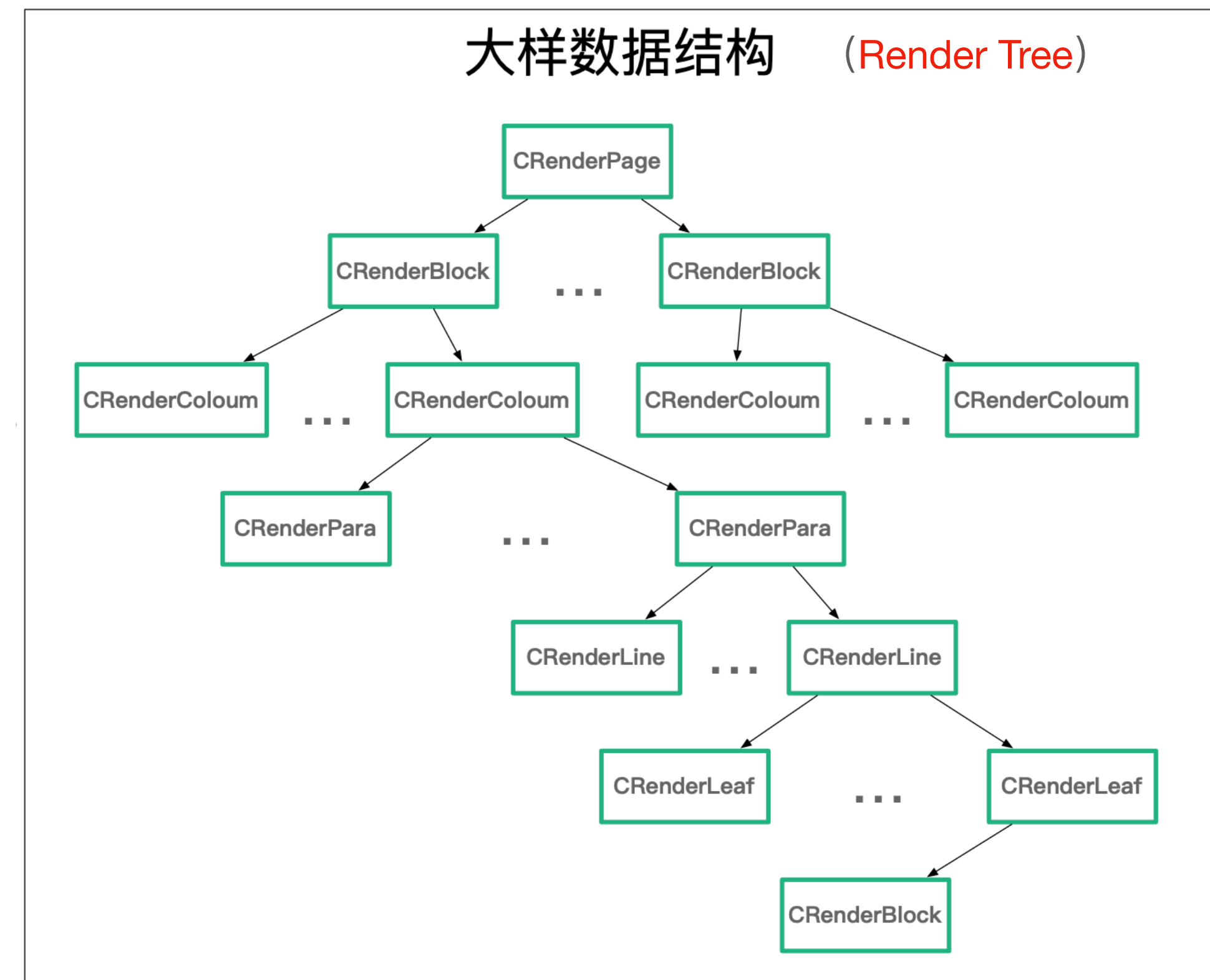
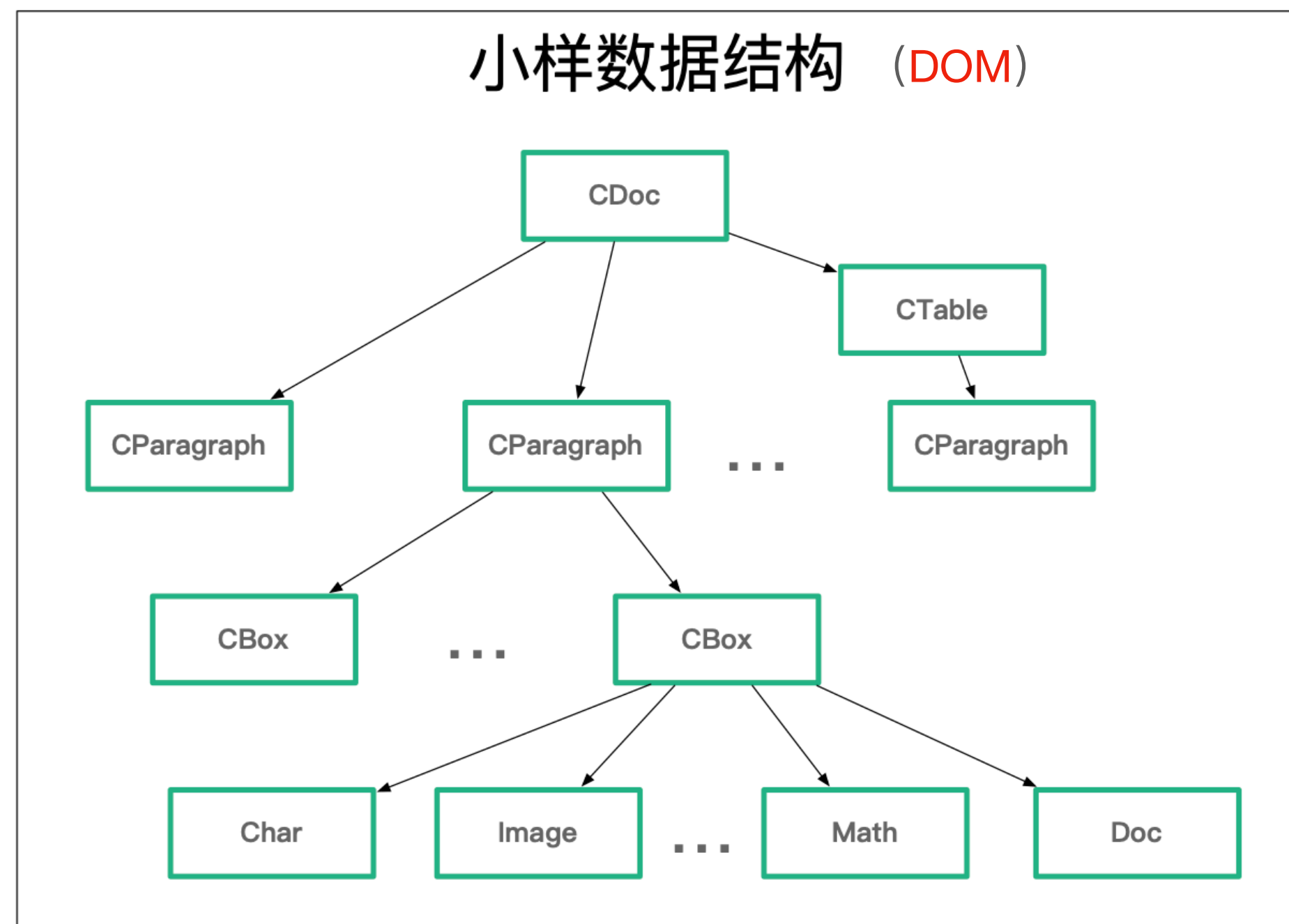


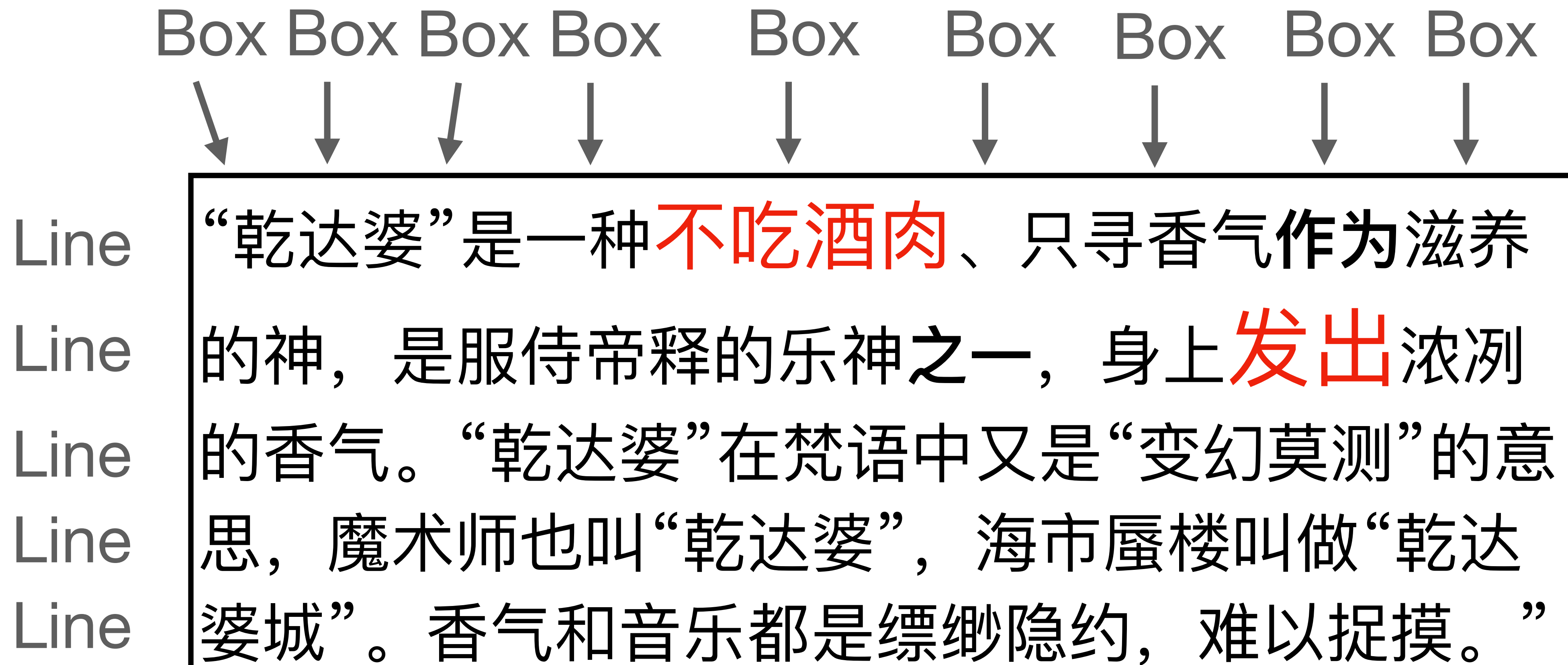
- 1、语言：采用 C++ 跨平台设计方案，一套排版引擎支撑多端
- 2、架构：从解析、排版、渲染全面自研
- 3、体验：提升排版能力和排版效果，提升阅读体验

整体架构图



T Chat 排版架构





T Chat 大样数据与小样数据



```
SDFNode(SDFNodeType node_type);  
~SDFNode();  
  
SDFNodeType node_type() { return m_node_type; };  
/* Node */  
void add_child(SDFNode *node);  
Int child_count() { return m_children.size(); };  
SDFNode* child_at(Int index);  
Int index() { return m_index; };  
  
static std::string string_node_type(SDFNodeType note_type);  
  
/* Property */  
void add_property(SDFProperty *v);  
void remove_property(const std::string &key);  
SDFProperty* get_property(const std::string &key);  
virtual std::string description();  
  
void set_element_name(std::string element_name) {  
    m_element_name = element_name; };  
std::string element_name() { return m_element_name; };
```

```
class BaseRender {  
public:  
    ~BaseRender();  
  
    bool start_draw(vector<shared_ptr<bd_reader_core::Meta>>&);  
    // Node  
    void add_child(BaseRender *);  
    bool has_children() { return !m_child_list.empty(); };  
    Int child_count() { return m_child_list.size(); };  
    BaseRender* child_at(const Int);  
    BaseRender* first_child();  
    BaseRender* last_child();  
    void insert_to_front(BaseRender *);  
    void remove_children(const Int from_index);  
    void remove_last_child();  
  
    void update_render_point(const Float x, const Float y);  
  
    void set_index(const Int index) { m_index = index; };  
    Int index() { return m_index; };  
  
#pragma mark -- Draw  
    shared_ptr<Meta> draw_text_in_rect(const Rect& rect, const  
        std::string& str, Font font, Int pn, Int start_word_index,  
        Int content_len);  
    shared_ptr<Meta> draw_image_in_rect(const Rect& rect, const  
        std::string& uri, const Int pn);  
  
#pragma mark -- Frame  
    /* frame */  
    void set_frame(const Rect &frame);  
    Rect frame() { return _frame; };  
    void set_x(const Float &x) { _frame.set_x(x); };
```

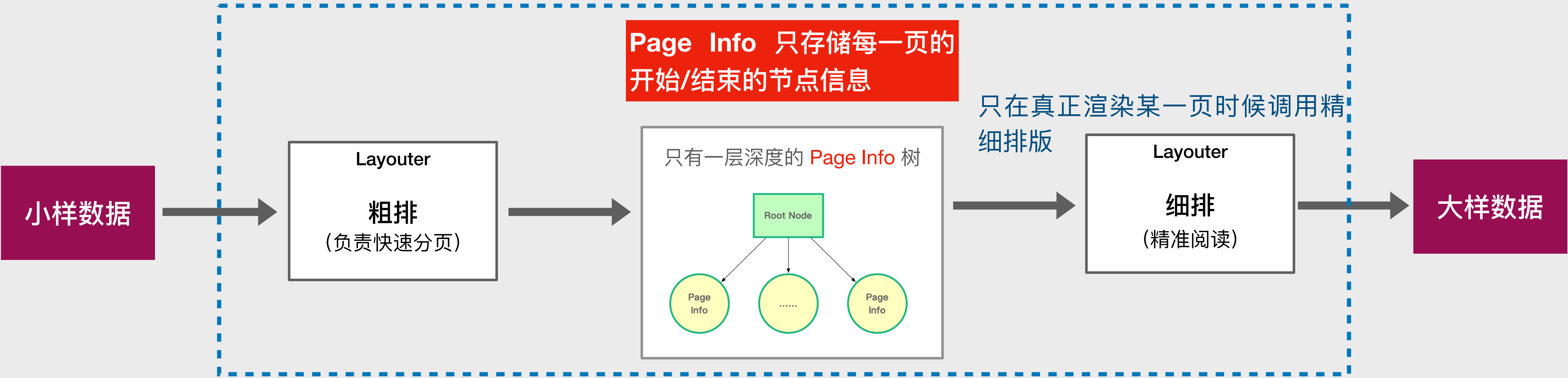
为什么是 Native 而不是 WebView?

T Chat 排版策略

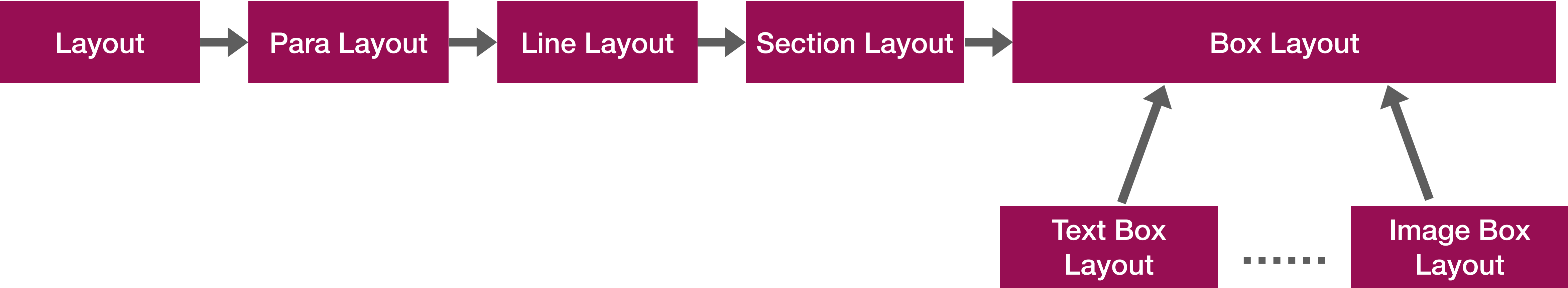


粗排策略： 只负责计算每一页能个排多少内容，而不进行详细的排版，如不会应用左右对齐算法等不影响排内容多少的策略，同时不存储大样数据。

细排策略： 应用全部排版策略，详细计算出每一个节点的位置坐标，完成渲染需要的全部数据。



```
// 分页
void start_paging(const Int &p_process_type, const Rect& p_layout_rect, std::vector<SdfPageInfo>&
    v_sdf_pages, const LayoutSpaceStruct &p_sp_stuct) override;
// sdf->ldf
shared_ptr<RenderPage> build_ldf(const SdfPageInfo &p_sdf, const Int &p_process_type, const Rect&
    p_layout_rect, const LayoutSpaceStruct &p_sp_stuct) override;
```

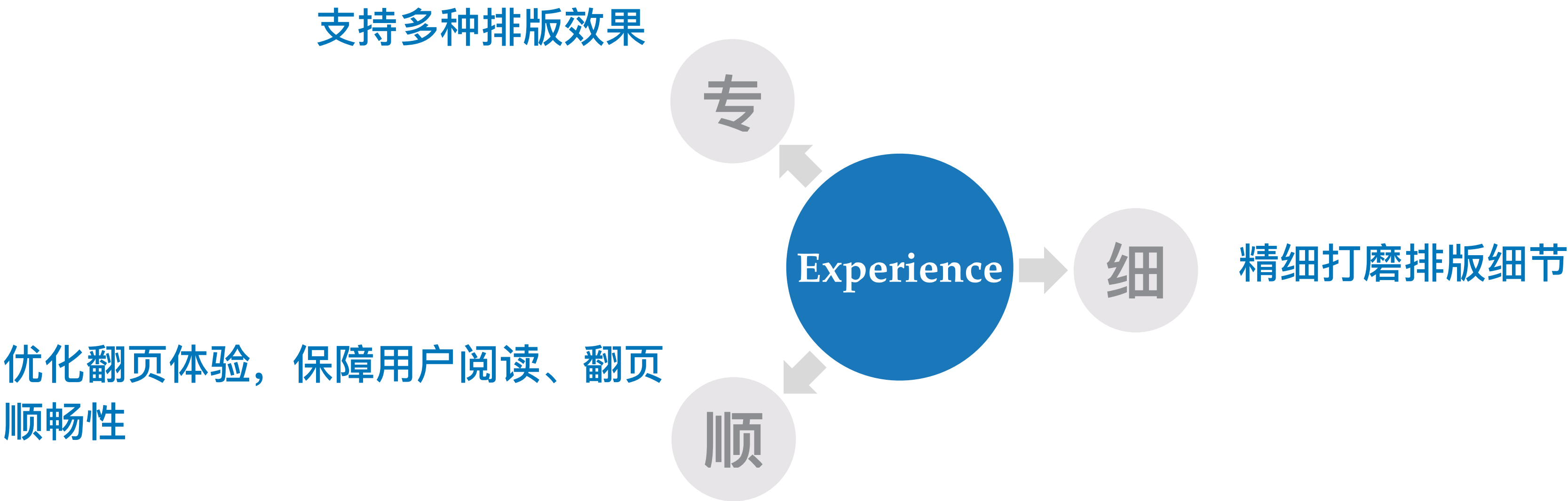
>Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum. Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum.

qián yán
前言

qǐng nǐ yī qǐ lái tǐ huì zhěng lǐ de mó fǎ
请你一起来体会整理的魔法

zhè běn shū jiǎng de shì zhǐ yào zhěng lǐ yī cì jiù jué
这本书讲的是“只要整理一次就绝对
duì bù huì huí fù zá luàn de zhěng lǐ fāng fǎ
不会对回复杂乱的整理方法”。

如何衡量排版体验效果是否出色？



小说来讲，保障用户连续阅读的连贯性，关键在于版面灰度分布均匀

即排版的字与字、字与标点、标点与标点等的空隙分布均匀，以达到版面的灰度均衡。灰度均衡的版面整体感更强，并最大程度保障读者阅读中不会被突然的空白打断，亦不会太紧密形成阅读的压迫感，整体提高阅读过程的连贯性)

行首全角标点造成上下段首不对齐

林渊主动打招呼。

【您好，宿主，恭喜您绑定

行首标点挤压

林渊主动打招呼。

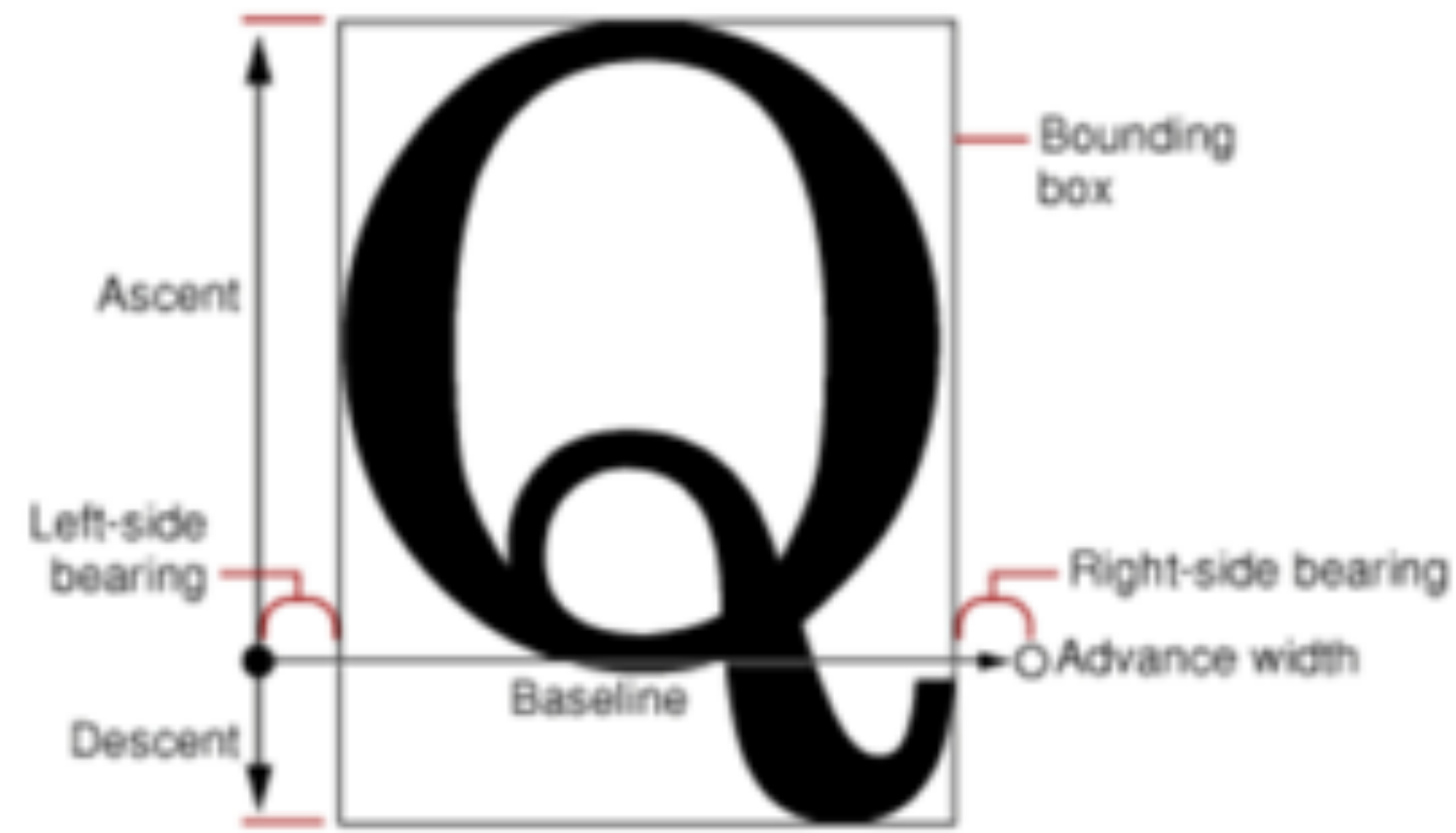
【您好，宿主，恭喜您绑定了

行内相邻标点造成空洞效果

没过多久，那机械般的电流
声再度于他的脑海中响起：【加

行内相邻标点挤压

没过多久，那机械般的电流
声再度于他的脑海中响起：【加载



Advance width

= glyph width + left bearing + right bearing

Ascent & Descent

相对基线的上下高度值，不同的字体可以有不同的 Ascent & Descent 值，常应用于像蒙文等上下排版的语言

Kerning

VA VI

منطقة باب السلام : فرع الجمعة - إسكاف الخلفي
ت - ٧٨٥٥٧٨ / ١١ - مدام - ٧٨٥٥٧٨ - ٢٢٢٤٥٧٨
مركز التجهيز - إسكاف الرشبي ت - ٥١٢٢١٦ - ١٠
محمد - ٧٧٦٤٠٨٢
فرع ع - ٧٨٥٠٨٨ / ٢ - ١٢ - شمس - ٧٧٤٠٥٦