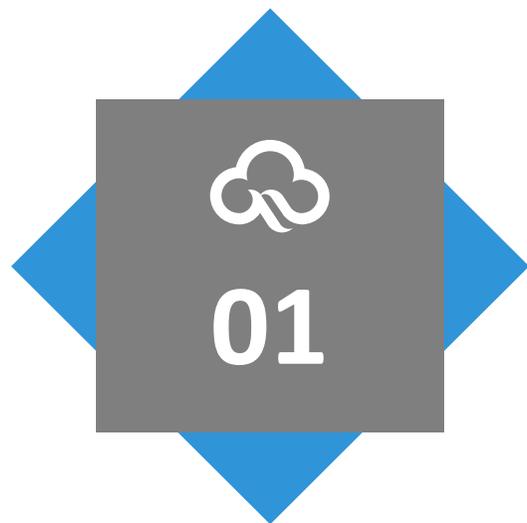


LiveVideoStackCon



# H.265金山云演进之路

金山云算法总监 朱政



## 序章：致敬x264



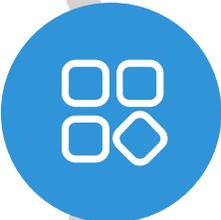


视频编码的守望者，  
最好的H.264编码器

创造了众所周知的工具和名  
词Mbtree , CRF , 2PASS



x264



繁荣了H.264的内容和生态

解放了视频编解码  
底层开发劳动力



## 除了不限速和最慢档，H.265软件的价值在哪里？

以对标x264各档次的速度为目标



眼前有景道不得  
崔颢题诗在上头



( 1080p, i5, 单线程, 1M~4M bps )

# 迄今达到的性能一览

性能摘要汇总

速度档次	vs. x264(RC=ABR)		vs. x265 (RC=ABR)		vs. VP9 (RC=VBR)	
	带宽节省	加速	带宽节省	加速	带宽节省	加速
superfast	43.7%	-5.4%	26.0%	212.6%	30.5%	147.9%
Veryfast	41.2%	8.9%	34.5%	123.2%	38.0%	75.5%
Slow	36.2%	-5.1%	23.7%	47.5%	32.6%	179.0%
Veryslow	35.3%	83.3%	10.9%	84.1%	23.1%	778.0%

veryfast为例性能摘要

ksc265 veryfast vs x264 veryfast						
Thread No.	Class	BDRATE-Y	BDRATE-U	BDRATE-V	BDRATE-YUV	Speed-Up
1	Class A	-43.6%	-41.7%	-45.0%	-43.7%	8.4%
	Class B	-51.0%	-55.6%	-56.4%	-51.9%	35.1%
	Class C	-38.2%	-49.8%	-49.5%	-40.3%	1.6%
	Class D	-31.5%	-51.5%	-50.8%	-35.1%	-15.2%
	ClassE	-48.0%	-61.5%	-63.9%	-51.7%	47.0%
	Game 30 fps	-36.4%	-43.7%	-42.5%	-38.0%	9.3%
	Average	-40.2%	-49.4%	-49.4%	-42.1%	13.1%
auto	Class A	-44.4%	-43.1%	-45.8%	-44.6%	4.2%
	Class B	-50.0%	-54.9%	-55.6%	-50.9%	26.9%
	Class C	-37.1%	-48.8%	-47.6%	-39.2%	-2.4%
	Class D	-30.2%	-50.0%	-50.1%	-33.8%	-0.8%
	ClassE	-46.4%	-60.1%	-62.2%	-50.1%	35.4%
	Game 30 fps	-35.5%	-42.8%	-41.4%	-37.1%	2.1%
	Average	-39.2%	-48.5%	-48.4%	-41.2%	8.9%



## 破茧而出



## 与x264复杂度对标

如何在保证HEVC高压缩比的情况下，速度和功耗无限接近x264

- 从零开始
- 从实时档次开始做





我们的2014

考虑从HM改版？

重构优化难驾驭

难以实现

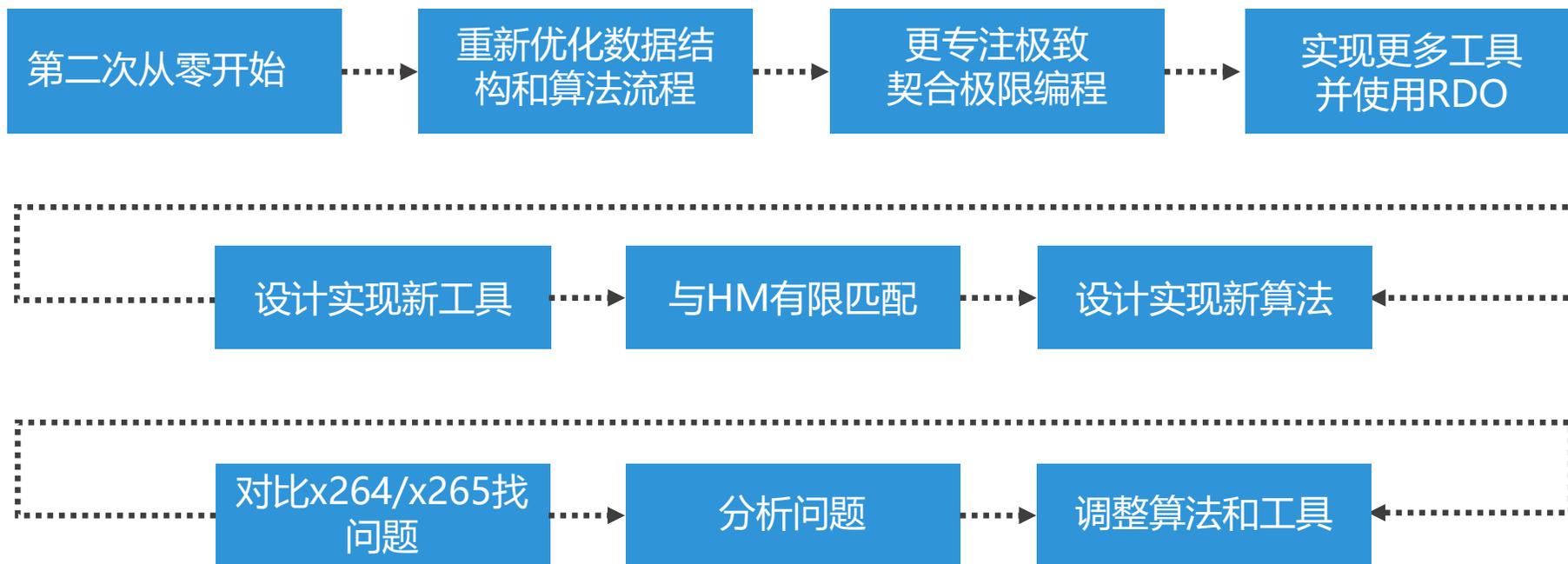
第一次从零开始

选取基础工具  
实现

效仿x264快速  
档，不做RDO

速度慢、RD差  
进退维谷

**从零开始做实时，随时随地重构，不惧推倒重来**



**积累之前的经验教训，对标准通篇更加熟悉**



快速迭代，不断试错，  
找到突破口



小团队结对编程，所有人  
熟悉所有代码



单元测试先行，严格  
codereview



每一处代码和数据结构  
精益求精



不怕麻烦，只怕性能不够  
**95%**的精力投入到**5%**的  
性能提升

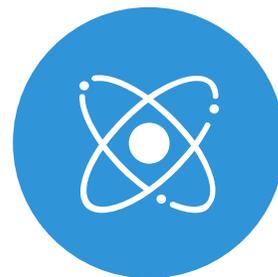


解码器追求极限速度



## SIMD优化

X86和ARM：ASM和intrinsics各一套  
变换和反变换实现无转置的方法



## C 代码优化

解码器完全无递归  
减少判断：编程过程CU边界仅两个  
Bool判断

作为编码器副产品打造，却耗费大量精力

比OpenHEVC平均提速2倍，全标准的支持，鲁棒性强

SEQS	Items	FFMPEG-KSC265 vs FFMPEG-OpenHEVC		
		iOS	android	解码提速
avantar	1920x1080	67.3 vs 24.0	126.7 vs 50.7	165%
BigShip	1280x720	113.0 vs 38.3	192.0 vs 84.3	161%
mobilephone	480x360	340.7 vs 157.7	572.3 vs 142.7	209%
mobile	832x480	285.3 vs 129.7	474.7 vs 89.3	276%
mobile	640x480	227.0 vs 96.7	384.7 vs 79.7	259%
( AVG )				<b>214%</b>

- 测试说明：KSC265解码器与openHEVC不同码率分辨率视频解码速度测试
- 测试设备：ipad min2(ios 7, apple A7), vivo xplay 5手机 ( android 5.1操作系统, 骁龙652 )



设定高目标，1月1个迭代



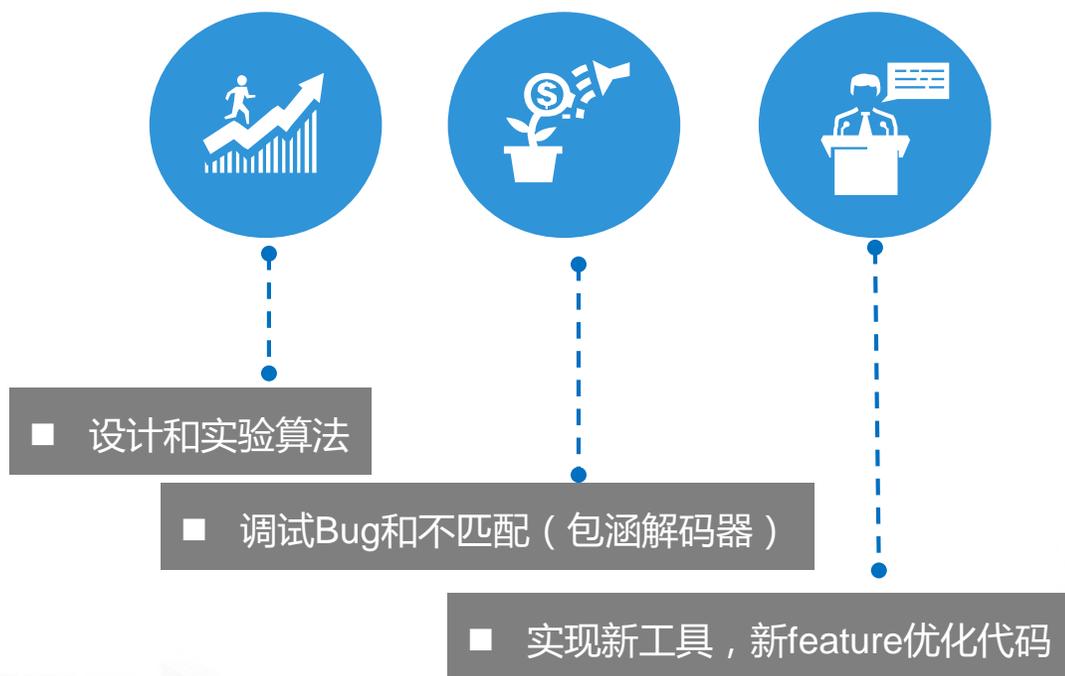
快速算法约**70**个，配置参数约**200**个



设计、实现、调试、反复试验

### 快速档次工具集

CU: 64x64 ~ 8x8	Skip/Merge	RDO
Intra4x4	IntraMode: 35	
deblock	SAO: BO/EO	
多参考帧	L0/L1/BI	1/4subpixel: 8P
预分析	scenecut	CUtree
signhide	TMVP	





- superfast5:1 -> 3:1 (speed up 3%, loss 1%)
- Veryfast: 10:1 -> 5:1
- ... 随着版本的成熟和算法数目增多，新算法收益越来越小
- Slow: 10:1
- Slower: 15:1
- Veryslows: 30:1

# 测试永不停歇

性能测试

正确性测试

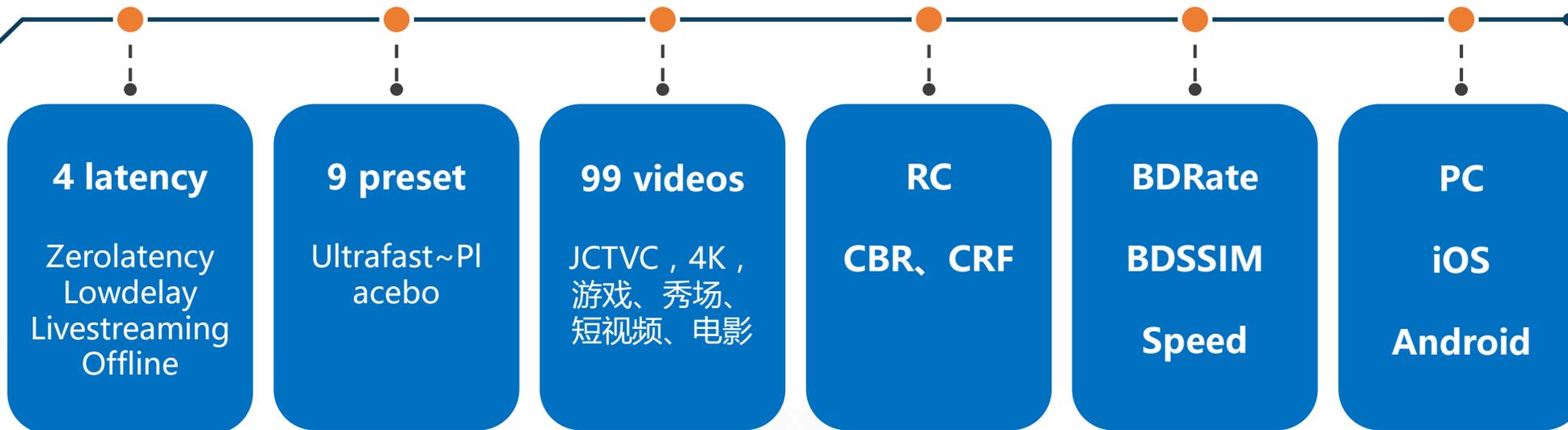
单元测试

CI测试

算法测试

版本迭代  
测试

## 性能测试



## 正确性测试

### 像素匹配

编码重建  
解码重建与HM解码重建像素匹配  
移动端MD5验证像素匹配

### 解码鲁棒性

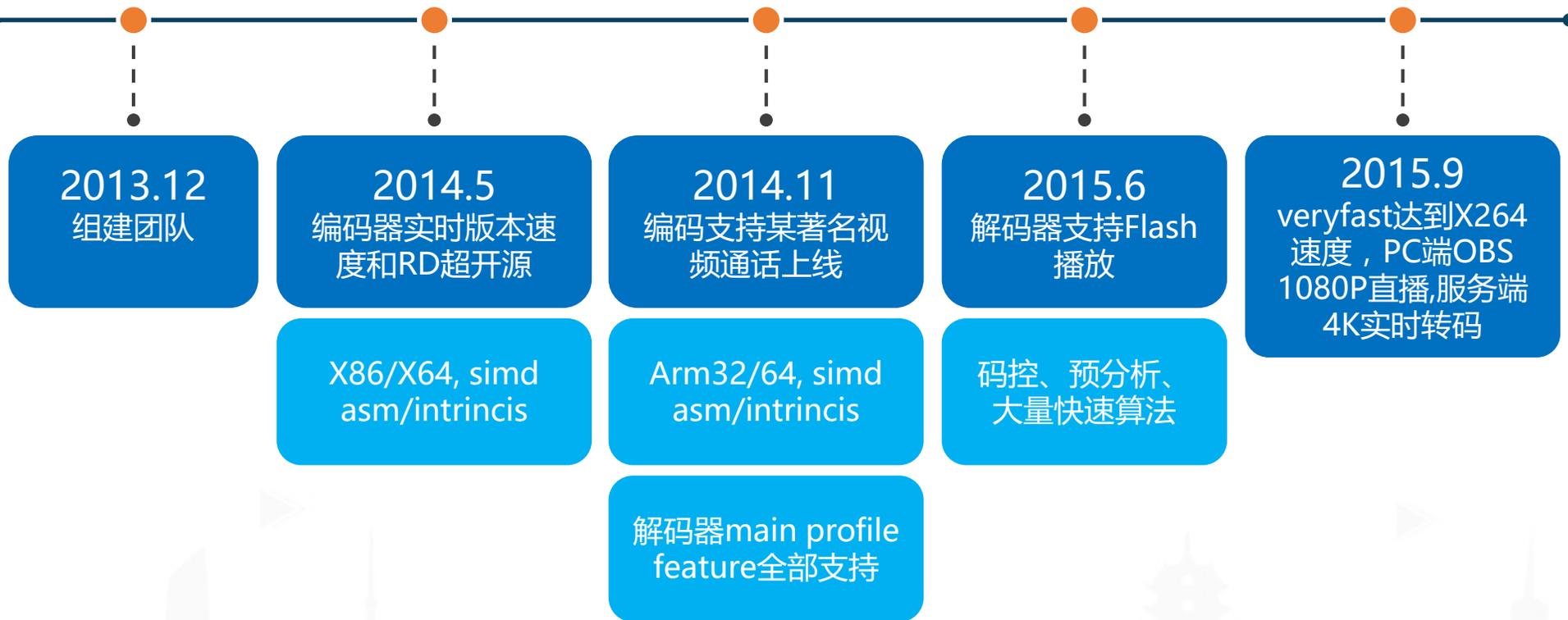
随机丢包  
随机更改Bit  
测试

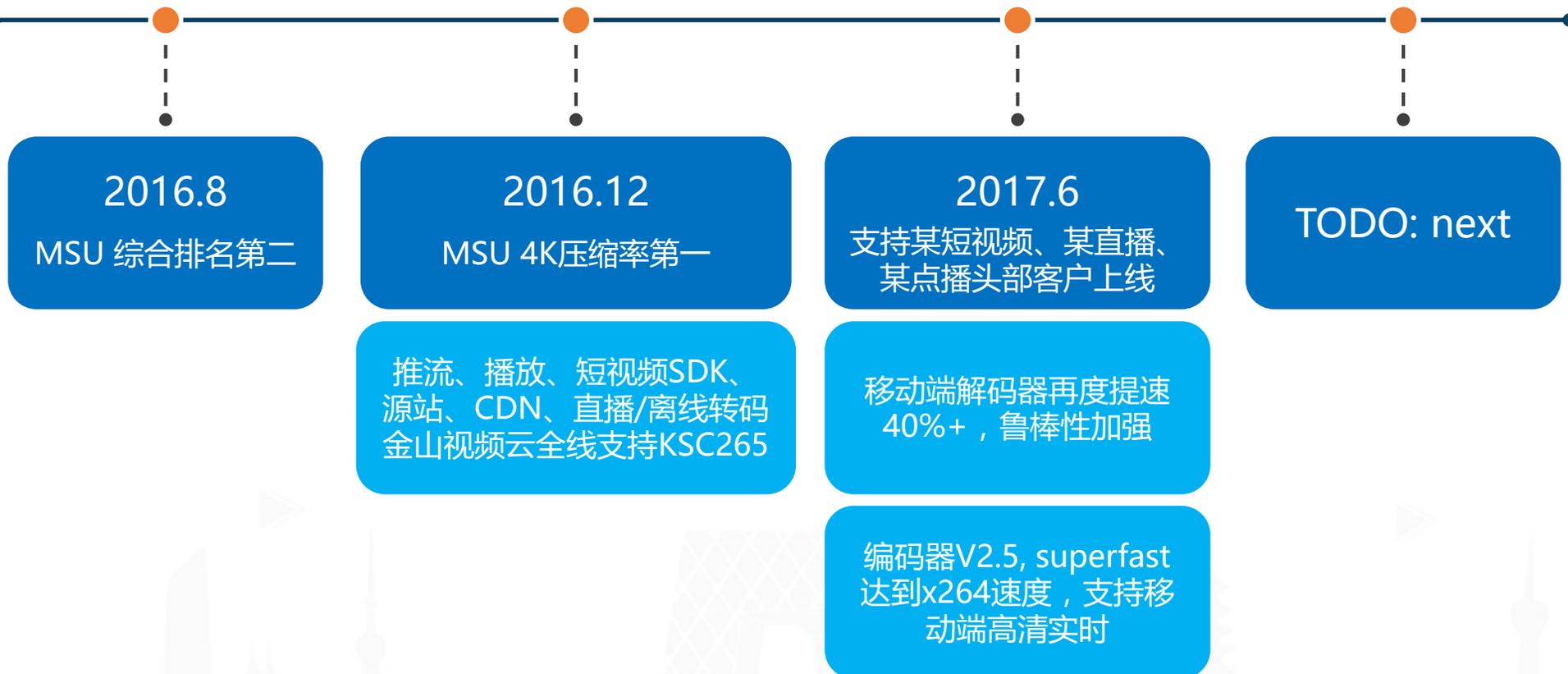
### 随机参数

所有可配  
参数随机

### 一致性

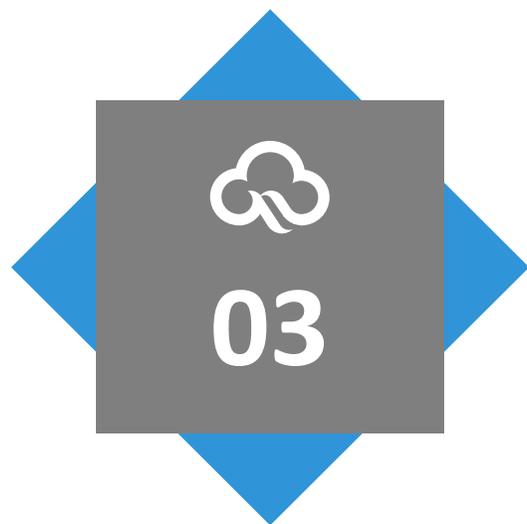
Windows、Linux、  
Mac、iOS、Andriod  
相同参数结果一致





- **Main profile, still profile**
- **Speed Control**
  - ultrafast ~ placebo
- **Latency Control**
  - Zerolatency, lowdelay, livestreaming, offline
- **Picture Type**
  - Intra period, scenecut, keyframe-request
  - RASL/close GOP
  - Max bframes, Adaptive bframes
  - Temporal scalable
  - Hierarchy P/B frames
- **Rate Control**
  - CQP, CRF, CBR, ABR
  - Vbv-bufsize, Vbv-maxrate, qpmax/qpmin
  - RateTolerance: stability of CBR
  - 2pass
  - Adaptive frame skipping
- **Parallel**
  - FPP, WPP, combination
  - Threadpool: task with priority

- **References**
  - Adaptive longterm
  - Adaptive multi-references
- **Quantization**
  - AQ, RDOQ, signhide, zerodetect
- **Algorithms**
  - Fast algorithms: 70+
  - Improve-quality algorithms
  - 200+ configurations on algorithms
- **Motion search**
  - DIA, HEX, improved UMH, EPZS
  - ¼ subpixel, several strength
- **Standard tools**
  - All sizes of CU/PU/TU, intra/inter, skip/merge
  - Deblock/SAO
  - Almost all...



## 痛点与解决之道



# ▶ 痛点和解决之道

困难	解决之道	解决指数
复杂度高	死磕、持续极致优化 复杂度对标x264	★★★★
浏览器Flash、 JS不支持	硬解白名单 H.264/H.265双备份 极致软解优化 H5播放探索	★★



用户痛点	解决之道
播放体验差	KSC265
带宽成本之痛	
用户追求高清内容	



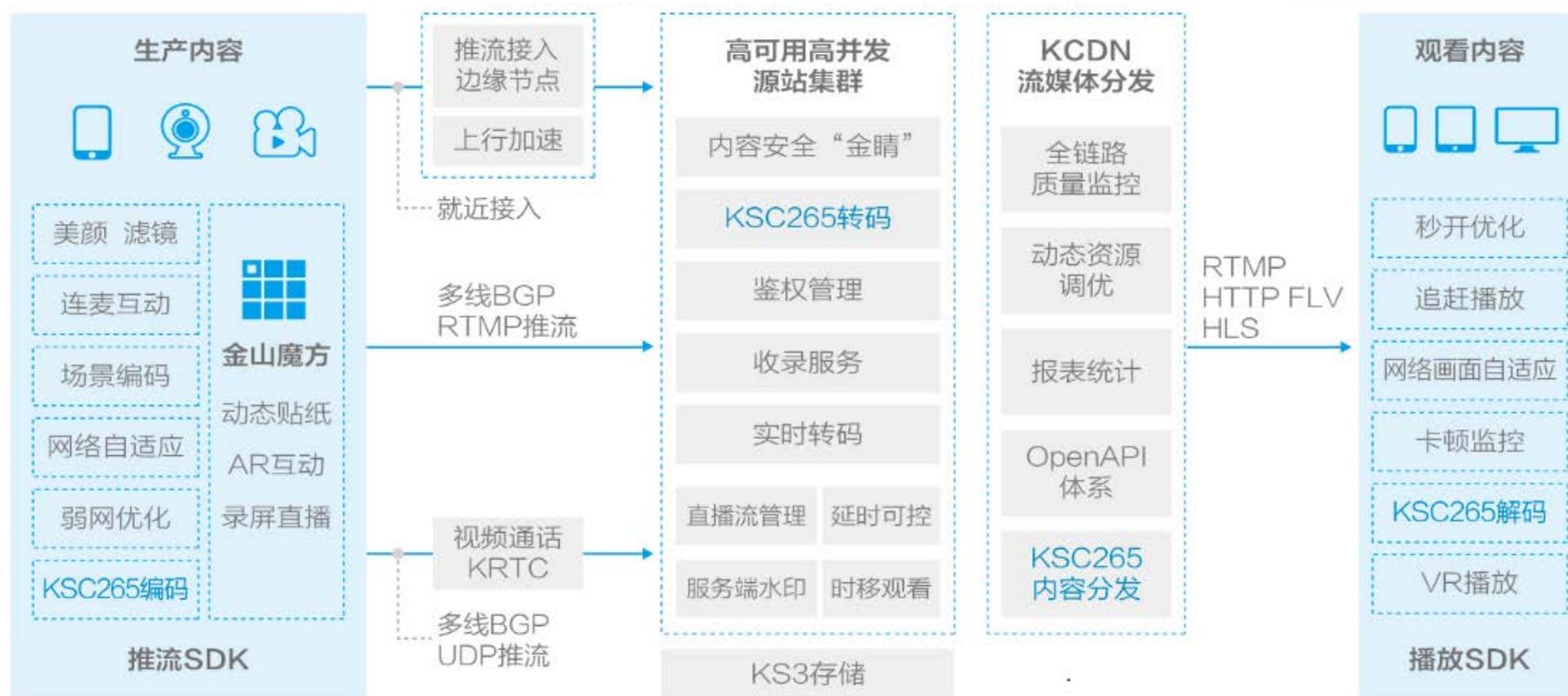
## 客户的疑虑

- 担心技术投入大
- 顾虑功耗和兼容性问题
- 担心有单点、无备份链路

## 解决之道

- 尽量与x264接口保持一致，Demo代码示范
- 客户原有的技术架构，最轻量化接入
- 白名单，软硬解切换
- iOS11全面支持硬编和软硬解
- 中低码率分辨率KSC265覆盖所有机型
- H.264/H.265双链路备份
- 软硬编软硬解切换
- 主流CDN厂商直播CDN都支持HEVC
- 与x265、OpenHEVC进行备份

# 金山视频云产品解决方案



云转码、客户端SDK，全面覆盖直播、点播、短视频解决方案

设置 KSC265编码器 帮助

浏览(.yuv)文件

1280x720\_15.yuv 确定

编码帧数:150  
编码速度:106.81 f/s  
压缩比:143

视频信息  
码率:1157.72 kbps  
分辨率:{1280, 720}  
帧率:15  
文件总时长:10.00 s

编码参数: ksc265enc -i 1280x720\_15.yuv -preset veryfast -latency offline -wdt 1280 -hgt 720 -fr 15 -threads 0 -br 1200 -b 1280x720\_15.yuv.265

编码时间:1.32 s  
编码帧数:150  
编码速度:113.21 f/s  
压缩比:141

视频信息  
码率:1170.46 kbps  
分辨率:{1280, 720}  
帧率:15  
文件总时长:10.00 s

设置 KSC265解码器 帮助

浏览(.265)文件

1280x720\_15.yuv.265 确定



解码参数: ksc265dec -b 1280x720\_15.yuv.265 -o 1280x720\_15.yuv.265.ksc.yuv -threads 0

分辨率: {1280, 720}  
渲染帧率: 0 (full speed)  
线程数: 0 (auto)  
解码时间: 2.50 s  
解码帧数: 150  
解码速度: 60.07

## 直观对比编解码压缩比与速度



iOS



安卓



## 未来展望



**实时档 -> 慢速档**  
**中低码率 -> 高码率**

**基础设施 -> 上层建筑**  
**码控、场景优化、自适应量化、主观调优**

**极致压缩：AI+视频编码**

# Thank You



扫码关注金山云LIVE

zhuzheng@Kingsoft.com

<https://github.com/ksvc/ks265codec>

LiveVideoStackCon



关注LiveVideoStack公众号

回复 **朱政** 为讲师评分