스마트폰을 오래쓸 수 있는 스마트한 솔루션

Dynamic Battery Manager Service

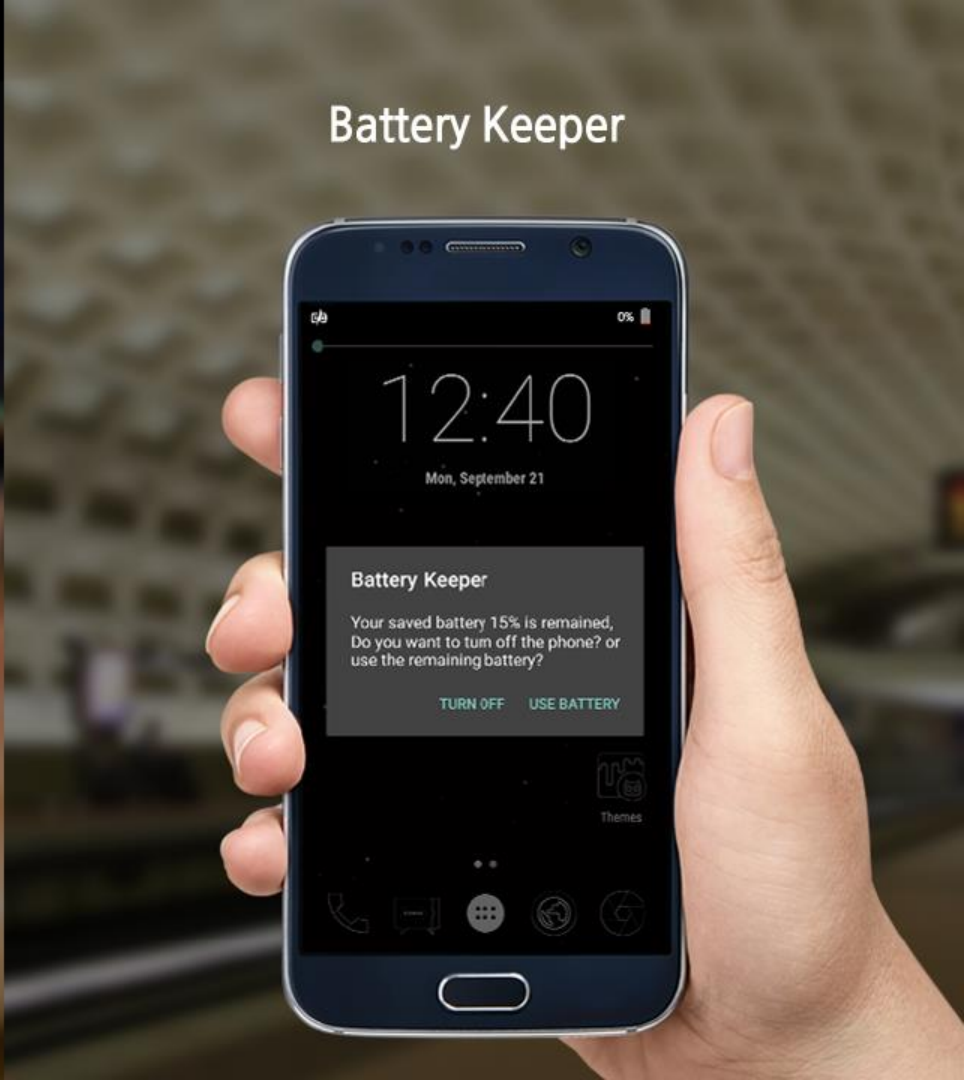
강신현 이정호
서보훈 황규정
김여진 이상은
박연주

저희는 스마트폰을 사용하는 모든 이들의
“배터리 고민”을 해결 해 보고자

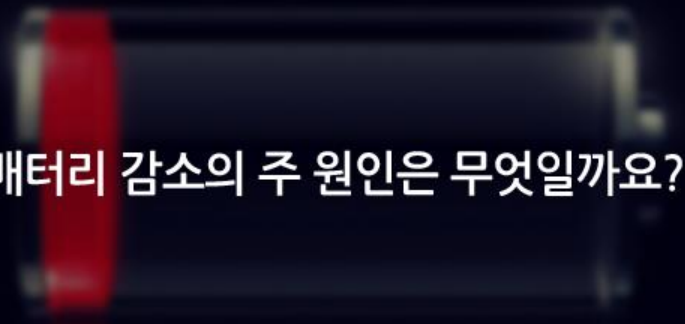
두 가지 솔루션을 만들어 보았습니다.

Dynamic Battery Saving Mode

Battery Keeper



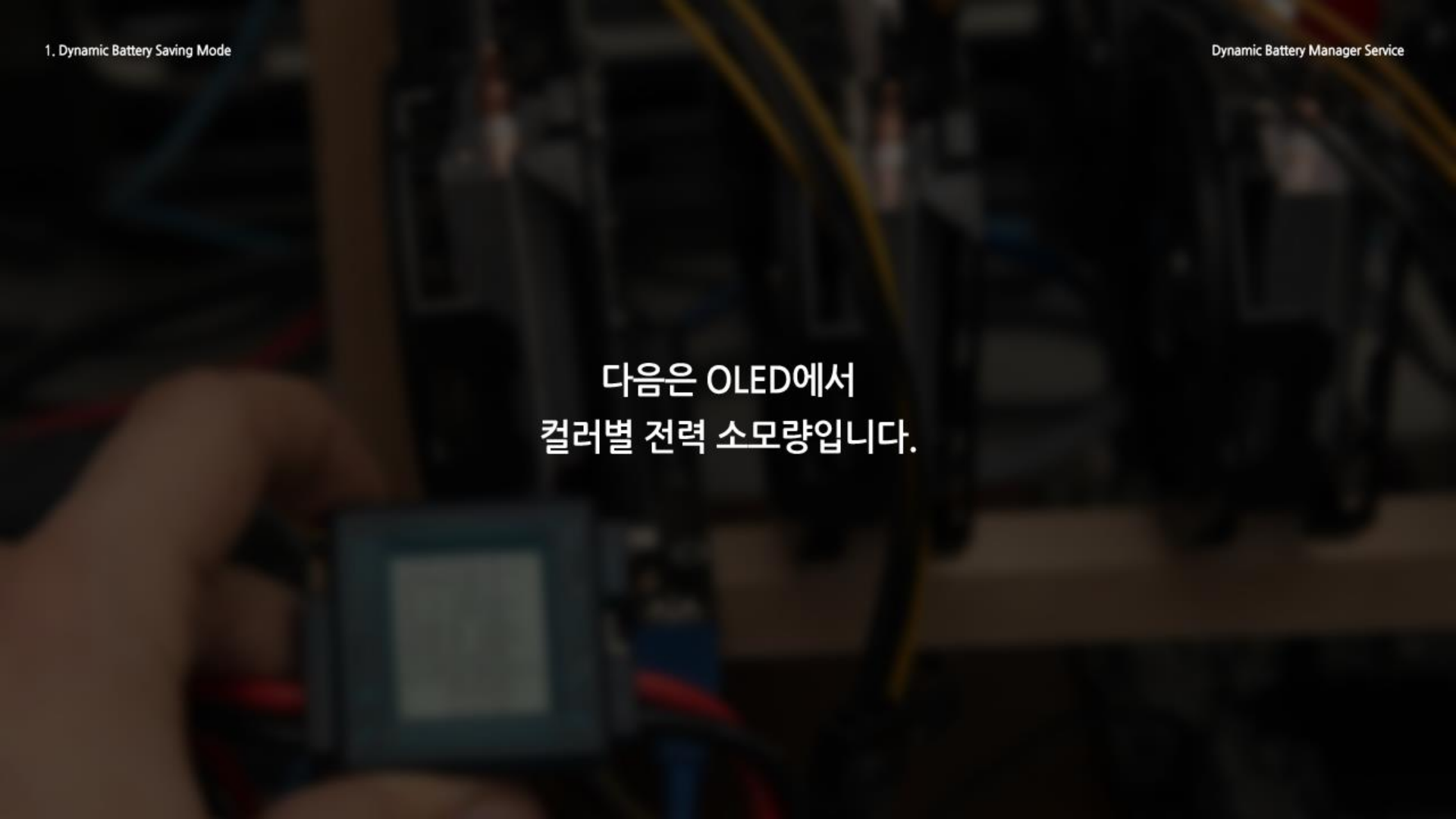
Dynamic Battery Saving Mode

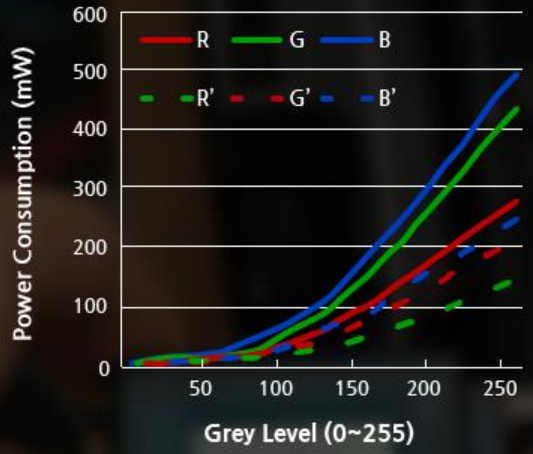


배터리 감소의 주 원인은 무엇일까요?


디스플레이, 즉 화면입니다.

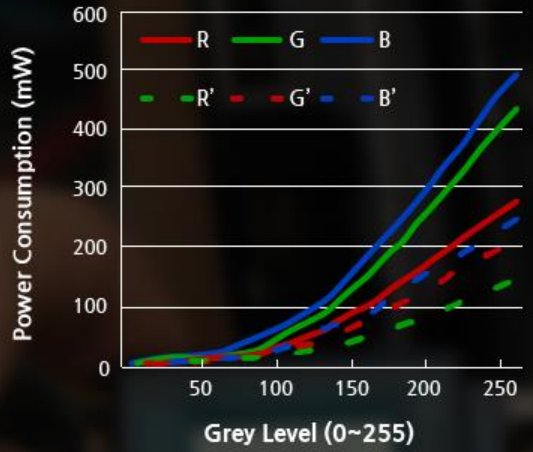

재미있는 사실은
컬러별 전력 소모량이 다르다는 것입니다.


다음은 OLED에서
컬러별 전력 소모량입니다.



Xiang Chen, 'How Energy Consumption in Smartphone Display Applications' 2013 논문 인용

R < G < B 순으로 전력소모가 크며,
BLACK은 서브픽셀을 끄게 됨으로
전력 소모를 일으키지 않습니다.



Xiang Chen, 'How Energy Consumption in Smartphone Display Applications' 2013 논문 인용

저희는 이러한 OLED 디스플레이의
특성에 착안하여 프로젝트를
진행했습니다.

Dynamic Battery Saving Mode 는
배터리 수준에 따라 배경화면을 변경하고
앱 화면의 컬러를 빼줌으로써 배터리 소모를 줄입니다

배터리 잔량이 100%~31% 일 경우

적용 시점- Default (30%, 15%, 5%)

배터리 잔량이 31%가 될 때 까지 절전 모드가 적용되지 않습니다.

배터리 잔량이 100%~31% 일 경우

적용 시점- Default (30%, 15%, 5%)

앱 화면 또한 마찬가지로 모든 컬러가 출력됩니다.

배터리 잔량이 30%~16% 일 경우

적용 시점- Default (30%, 15%, 5%)

배터리 잔량이 30%가 되면 아이콘의 겉 테두리에 흰색이 씌워져 구분됩니다.

배터리 잔량이 30%~16% 일 경우

적용 시점- Default (30%, 15%, 5%)

앱 화면은 R - 150 , G - 135 , B - 135 로 화면 서브 픽셀의 비율을 감소 시킵니다.

배터리 잔량이 15%~6% 일 경우

적용 시점- Default (30%, 15%, 5%)

배터리 잔량이 15%가 되면 아이콘이 아이덴티티 컬러의 Line 형태로 표현됩니다.

배터리 잔량이 15%~6% 일 경우

적용 시점- Default (30%, 15%, 5%)

앱 화면은 R - 85 , G - 75 , B - 75 로 화면 서브 픽셀의 비율을 감소 시킵니다.

배터리 잔량이 5%~0% 일 경우

적용 시점- Default (30%, 15%, 5%)

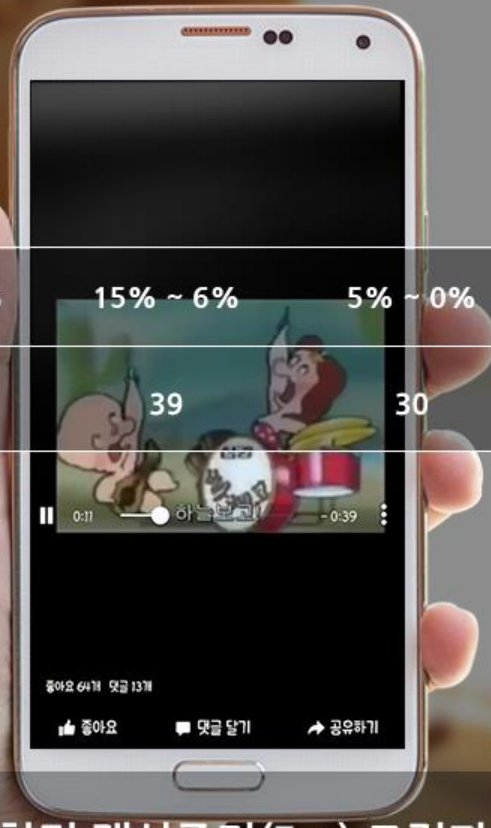
배터리 잔량이 5%가 되면 바탕화면에 그레이스케일이 적용됩니다.

배터리 잔량이 5%~0% 일 경우

적용 시점- Default (30%, 15%, 5%)

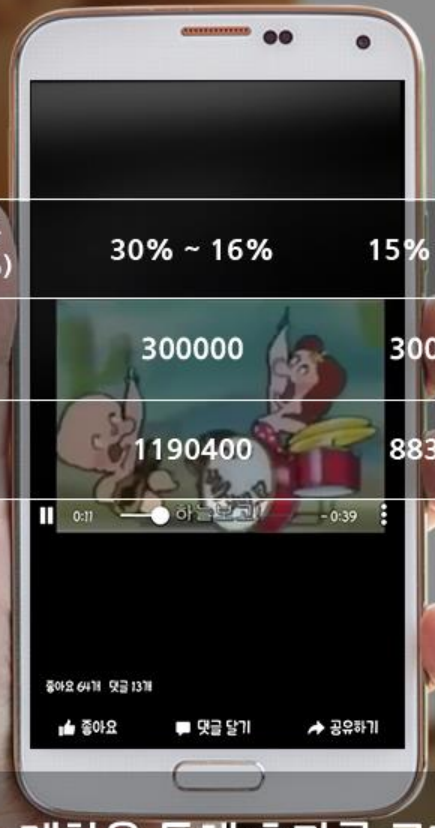
앱 화면에 그레이스케일이 적용됩니다.

Battery	30% ~ 16%	15% ~ 6%	5% ~ 0%	비고
Fps	48	39	30	동영상 인지 시 30

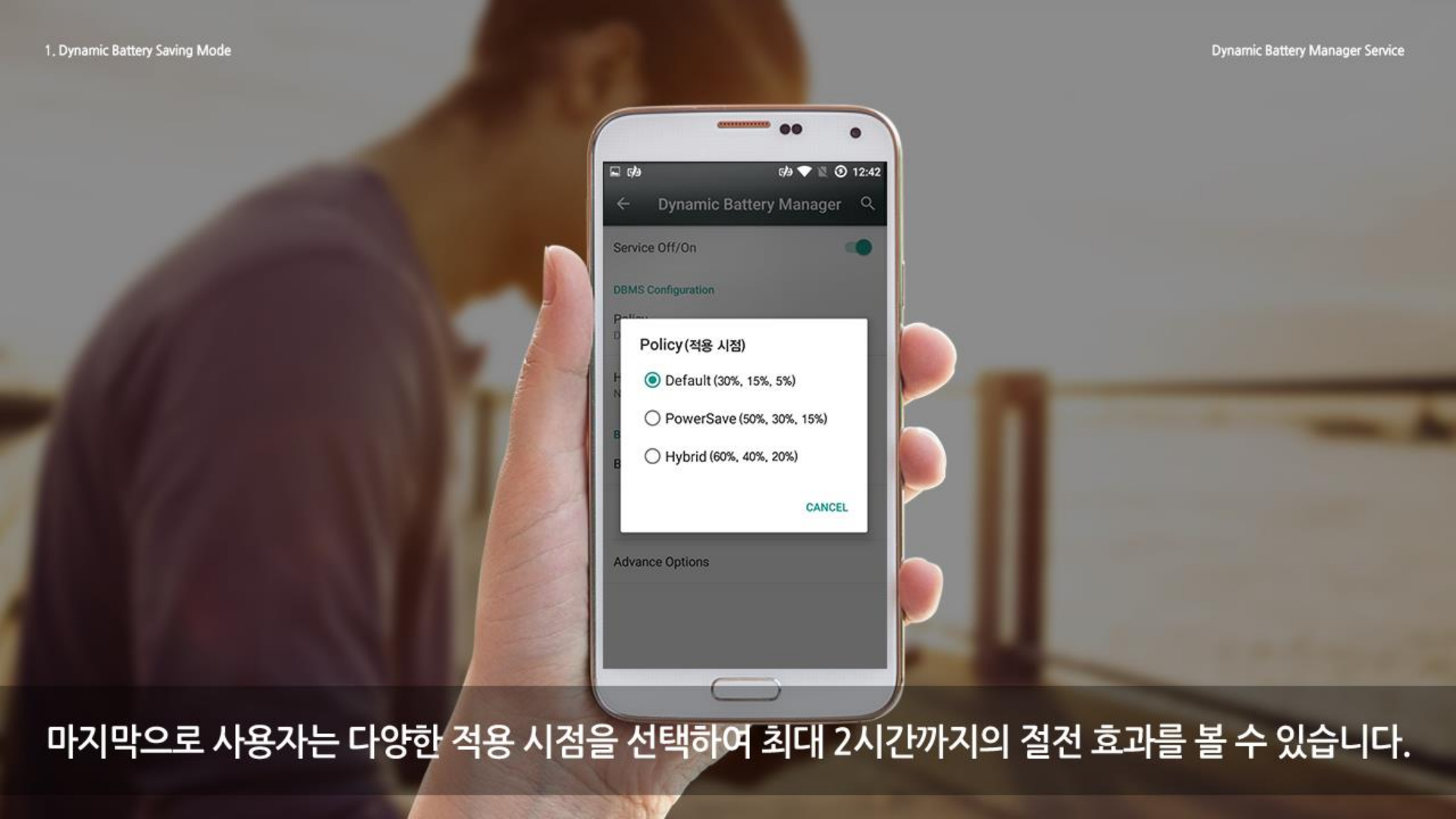


또한 화면 갱신주기(Fps) 조절과

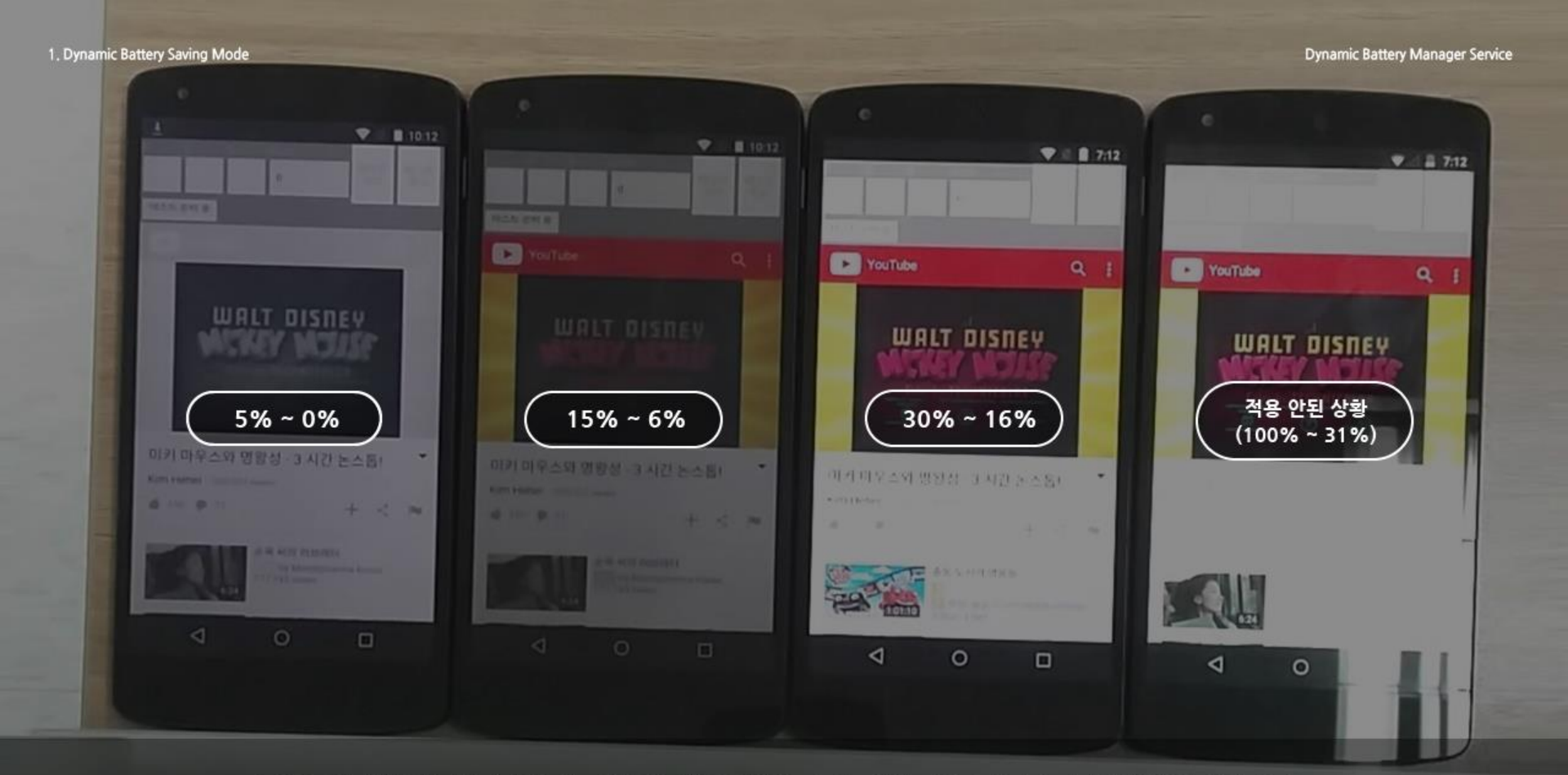
Battery	적용 안된 상황 (100% ~ 31%)	30% ~ 16%	15% ~ 6%	5% ~ 0%
CPU MIN	1267200	300000	300000	300000
CPU MAX	2419200	1190400	883200	652800



CPU 최대 클럭 주파수 제한을 통해 효과를 극대화할 수 있습니다.



마지막으로 사용자는 다양한 적용 시점을 선택하여 최대 2시간까지의 절전 효과를 볼 수 있습니다.



5%의 배터리를 몇 초 만에 사용하는 지 성능을 테스트 해 보았습니다.

Battery	적용 안된 상황	30% ~ 16%	15% ~ 6%	5% ~ 0%
5%당 소모시간 (초)	800~900초	1000~1200초	1300~1500초	1500~1600초
미적용 상황과 차이 (분)		+ 3~6분	+ 8분~11분	+ 11분~13분

5%당 최대 11분~ 13분 배터리 효율 향상을 보였습니다.

Battery	적용 안된 상황	30% ~ 16%	15% ~ 6%	5% ~ 0%
5%당 소모시간 (초)	800~900초	1000~1200초	1300~1500초	1500~1600초
미적용 상황과 차이 (분)		+ 3~6분	+ 8분~11분	+ 11분~13분

즉, 30%부터 적용된다면 사용자는 약 44분~64분동안 배터리를 더 사용할 수 있습니다.

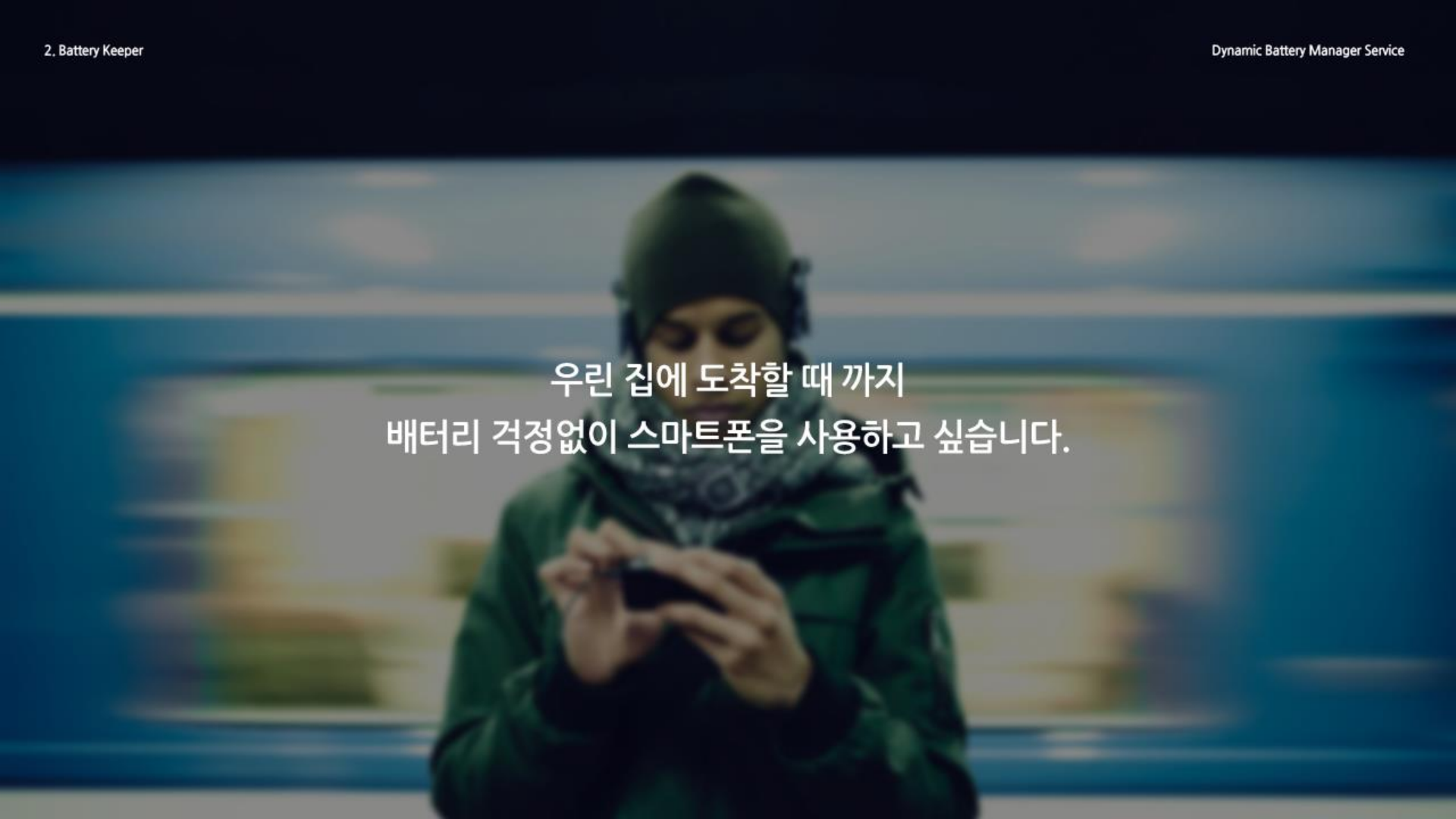
	배터리 절약 시간	비고
Default mode	약 31분 ~ 53분	테스트별 오차범위 5%마다 ±100초씩 변화율을 가짐
PowerSave mode	약 1시간 ~ 2시간	
Hybrid mode	약 1시간 ~ 2시간	

모드 별 기대되는 절약시간 입니다.

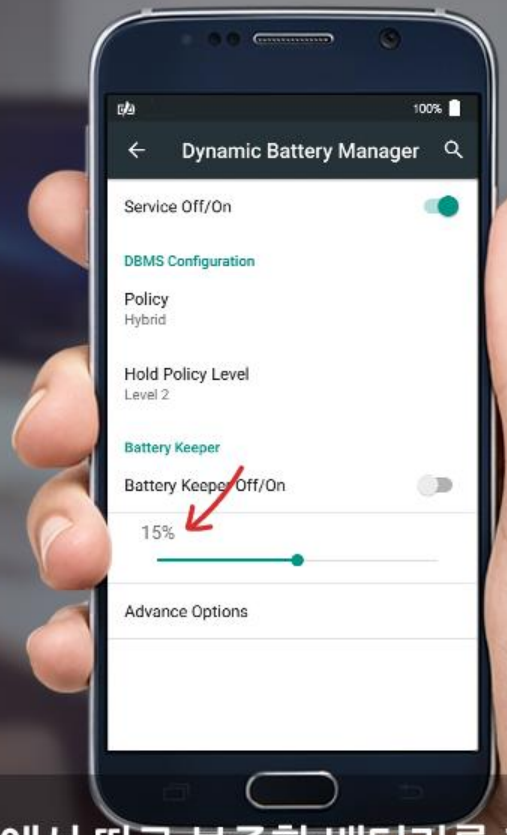
Dynamic Battery Saving Mode를 통해
사용자는 기존의 절전모드보다 향상된
사용성 및 절전효과를 체감 할 수 있습니다.

Battery Keeper

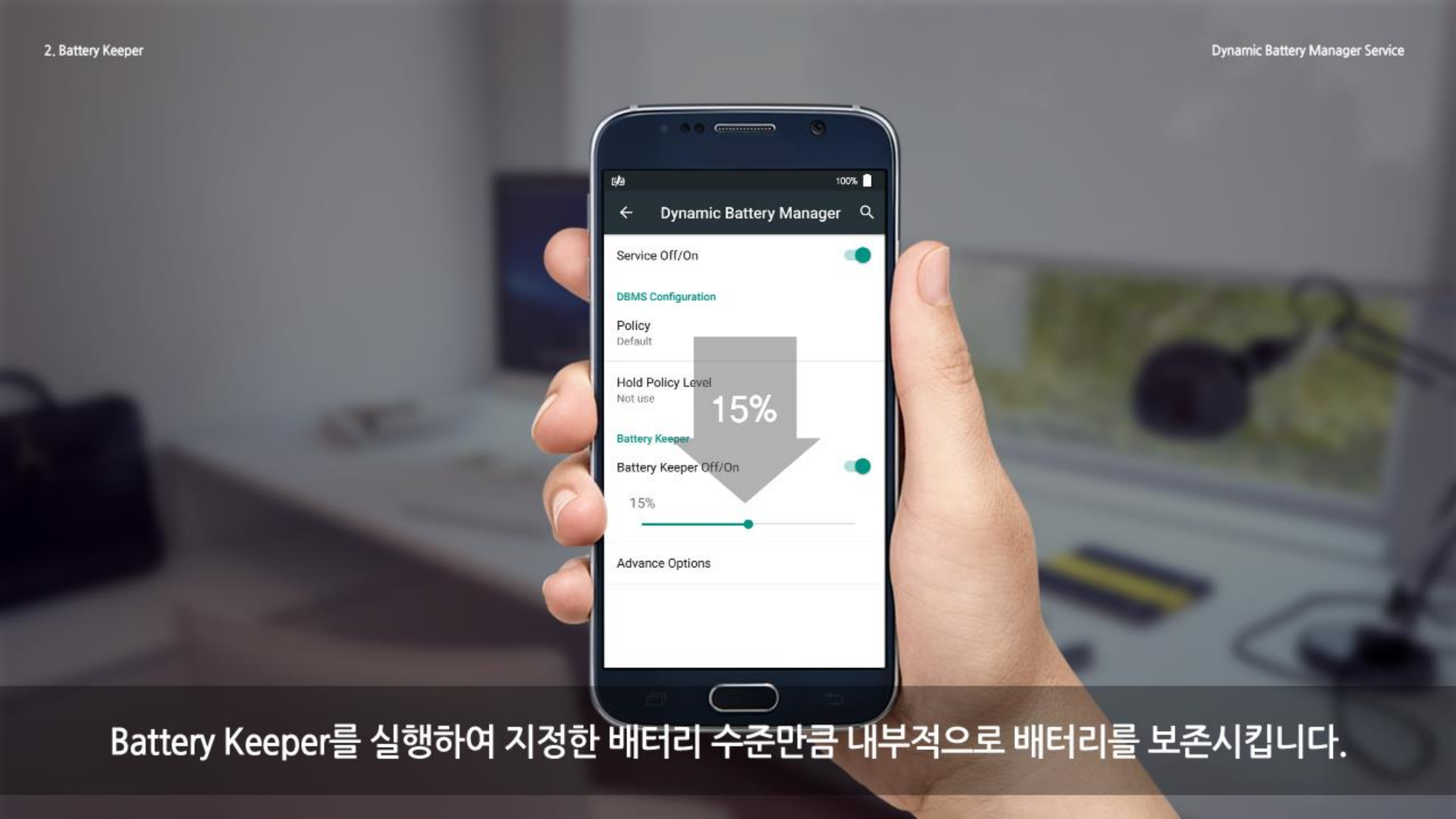

귀가할 때 쯤이면 항상 부족한 배터리


우린 집에 도착할 때 까지
배터리 걱정없이 스마트폰을 사용하고 싶습니다.

Battery Keeper 는
필요 상황에 마음껏 스마트폰을 쓸 수 있도록
배터리를 지정하여 따로 보존 해 줍니다.



먼저 설정에서 따로 보존할 배터리를 지정합니다.



Battery Keeper를 실행하여 지정한 배터리 수준만큼 내부적으로 배터리를 보존시킵니다.

사용자가 하루동안 스마트폰을 사용하여

배터리가 0%가 되면 저장해 둔 배터리를 사용할 것인지 여부를 묻는 팝업이 보입니다.

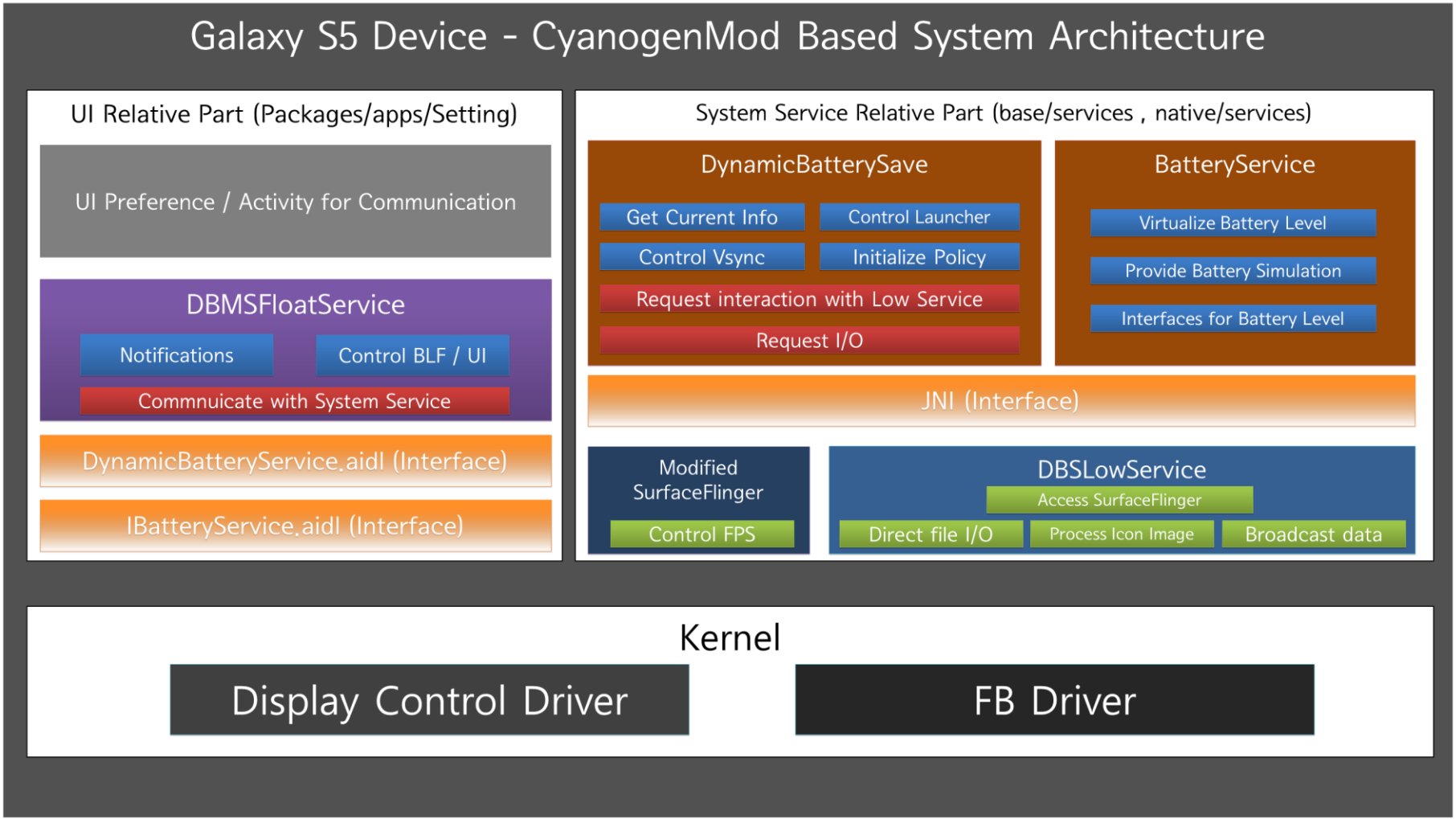
TURN OFF버튼을 누르면 스마트폰의 전원이 종료되며,

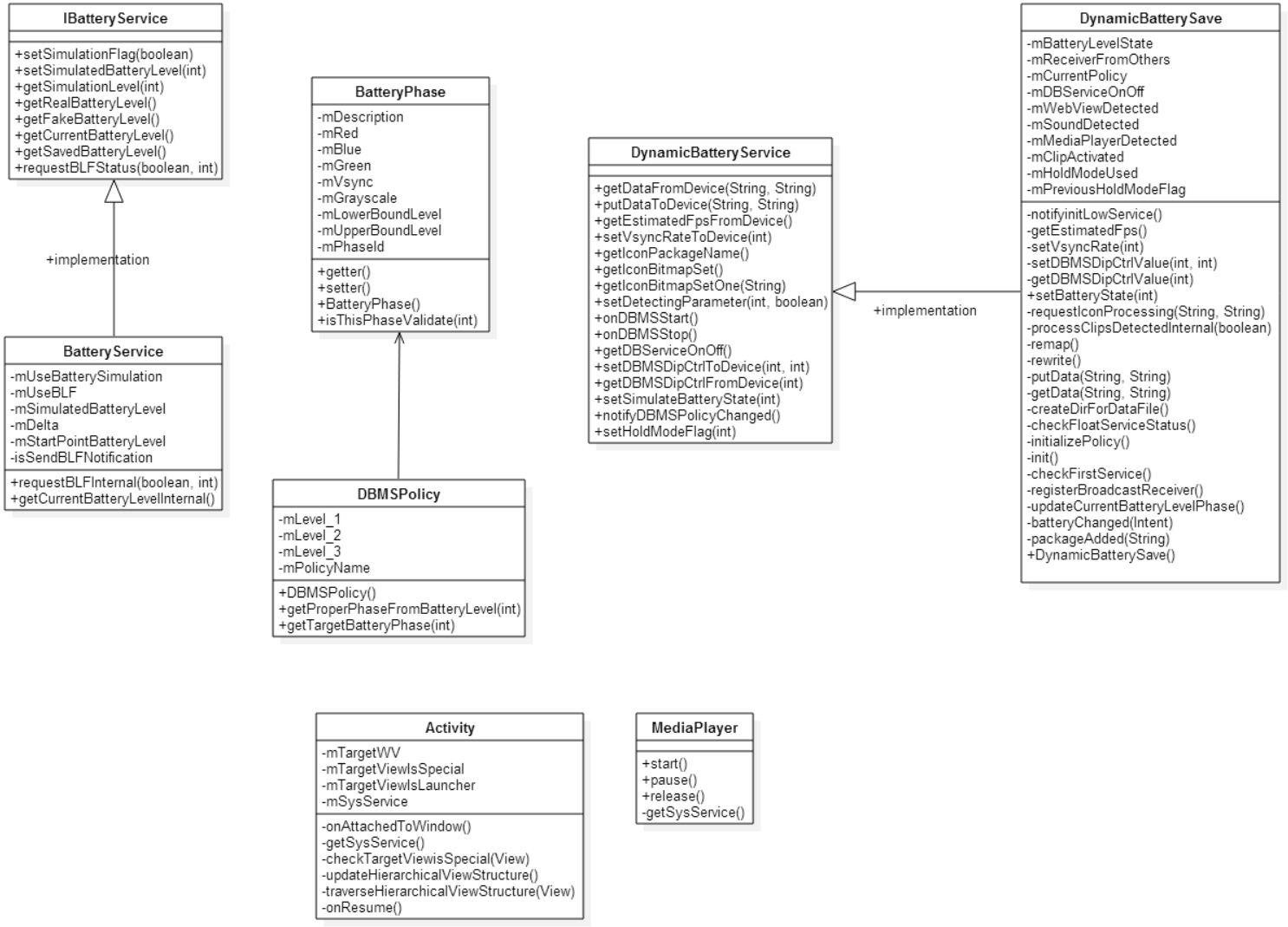
반대로 USE BATTERY버튼을 누르면 15%의 배터리가 채워집니다.

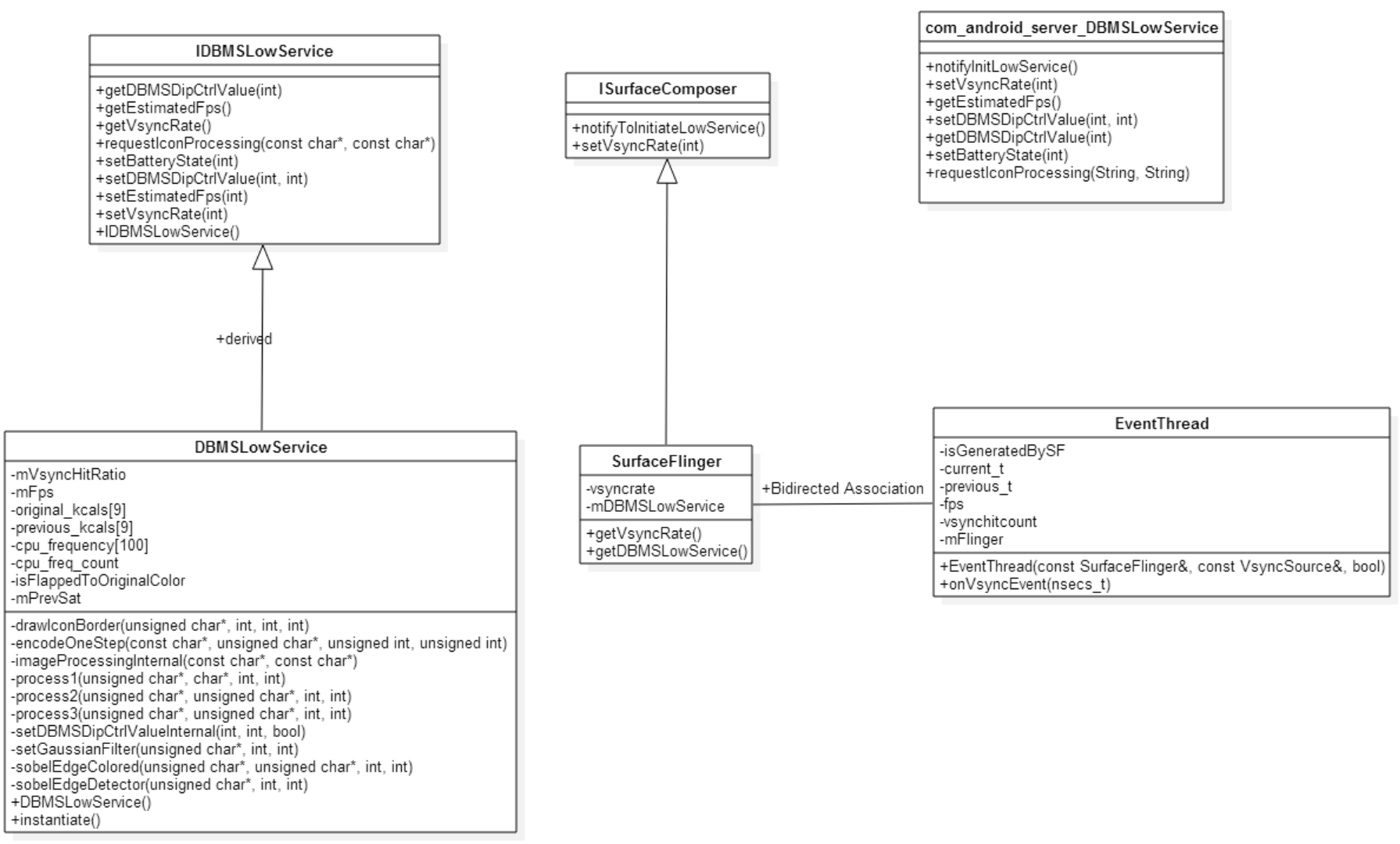
Battery Keeper를 통해
사용자는 기존 절전모드의 개념을 바꾼
새로운 방식으로 배터리를 아낄 수 있습니다.

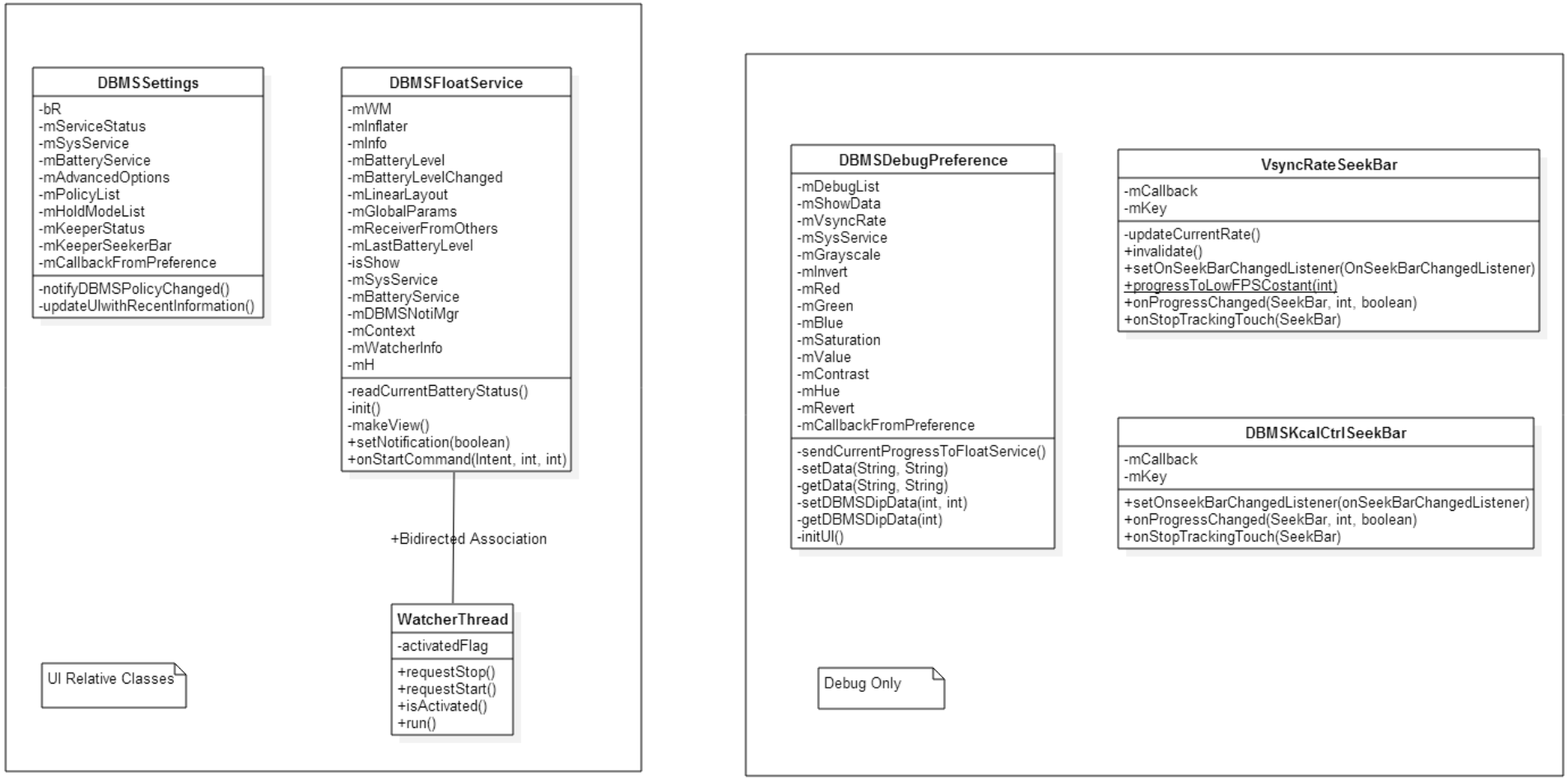
당신의 스마트폰을 더욱 오래 쓸 수 있는 새로운 절전 솔루션

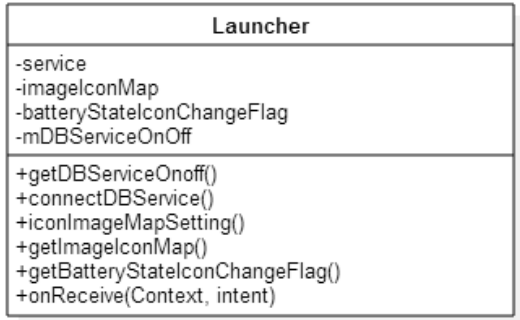
Dynamic Battery Manager Service

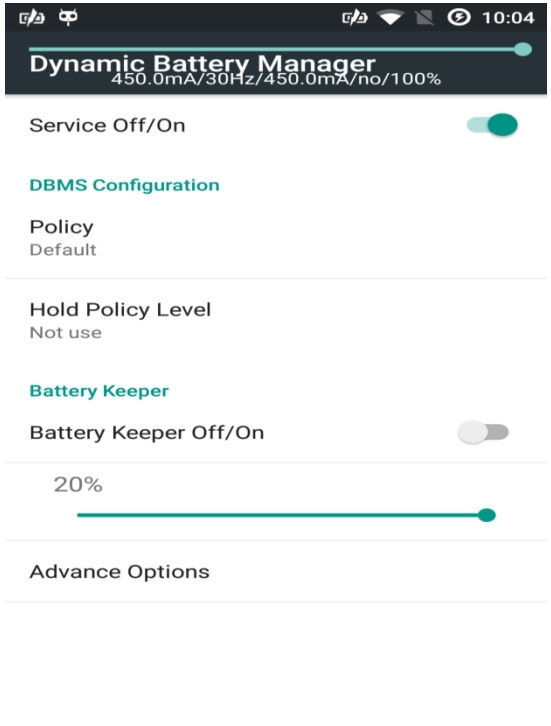

Galaxy S5 Device - CyanogenMod Based System Architecture
UI Relative Part (Packages/apps/Setting)
UI Preference / Activity for Communication
DBMSFloatService
Notifications
Control BLF / UI
Commnuicate with System Service
DynamicBatteryService.aidl (Interface)
IBatteryService.aidl (Interface)
System Service Relative Part (base/services , native/services)
DynamicBatterySave
Get Current Info
Control Launcher
Control Vsync
Initialize Policy
Request interaction with Low Service
Request I/O
BatteryService
Virtualize Battery Level
Provide Battery Simulation
Interfaces for Battery Level
JNI (Interface)
Modified SurfaceFlinger
Control FPS
DBSLowService
Access SurfaceFlinger
Direct file I/O
Process Icon Image
Broadcast data
Kernel
Display Control Driver
FB Driver

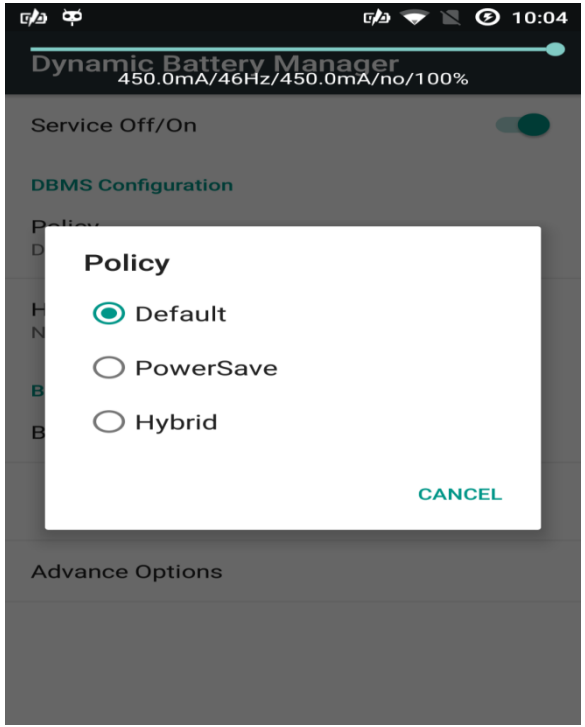

IBatteryService
+setSimulationFlag(boolean)
+setSimulatedBatteryLevel(int)
+getSimulationLevel(int)
+getRealBatteryLevel()
+getFakeBatteryLevel()
+getCurrentBatteryLevel()
+getSavedBatteryLevel()
+requestBLFStatus(boolean, int)
+implementation
BatteryService
-mUseBatterySimulation
-mUseBLF
-mSimulatedBatteryLevel
-mDelta
-mStartPointBatteryLevel
-isSendBLFNotification
+requestBLFInternal(boolean, int)
+getCurrentBatteryLevelInternal()
BatteryPhase
-mDescription
-mRed
-mBlue
-mGreen
-mVsync
-mGrayscale
-mLowerBoundLevel
-mUpperBoundLevel
-mPhaseId
+getter()
+setter()
+BatteryPhase()
+isThisPhaseValidate(int)
DBMSPolicy
-mLevel_1
-mLevel_2
-mLevel_3
-mPolicyName
+DBMSPolicy()
+getProperPhaseFromBatteryLevel(int)
+getTargetBatteryPhase(int)
DynamicBatteryService
+getDataFromDevice(String, String)
+putDataToDevice(String, String)
+getEstimatedFpsFromDevice()
+setVsyncRateToDevice(int)
+getIconPackageName()
+getIconBitmapSet()
+getIconBitmapSetOne(String)
+setDetectingParameter(int, boolean)
+onDBMSStart()
+onDBMSStop()
+getDBServiceOnOff()
+setDBMSDipCtrlToDevice(int, int)
+getDBMSDipCtrlFromDevice(int)
+setSimulateBatteryState(int)
+notifyDBMSPolicyChanged()
+setHoldModeFlag(int)
+implementation
DynamicBatterySave
-mBatteryLevelState
-mReceiverFromOthers
-mCurrentPolicy
-mDBServiceOnOff
-mWebViewDetected
-mSoundDetected
-mMediaPlayerDetected
-mClipActivated
-mHoldModeUsed
-mPreviousHoldModeFlag
-notifyinitLowService()
-getEstimatedFps()
-setVsyncRate(int)
-setDBMSDipCtrlValue(int, int)
-getDBMSDipCtrlValue(int)
+setBatteryState(int)
-requestIconProcessing(String, String)
-processClipsDetectedInternal(boolean)
-remap()
-rewrite()
-putData(String, String)
-getData(String, String)
-createDirForDataFile()
-checkFloatServiceStatus()
-initializePolicy()
-init()
-checkFirstService()
-registerBroadcastReceiver()
-updateCurrentBatteryLevelPhase()
-batteryChanged(Intent)
-packageAdded(String)
+DynamicBatterySave()
Activity
-mTargetWV
-mTargetViewIsSpecial
-mTargetViewIsLauncher
-mSysService
-onAttachedToWindow()
-getSysService()
-checkTargetViewisSpecial(View)
-updateHierarchicalViewStructure()
-traverseHierarchicalViewStructure(View)
-onResume()
MediaPlayer
+start()
+pause()
+release()
-getSysService()

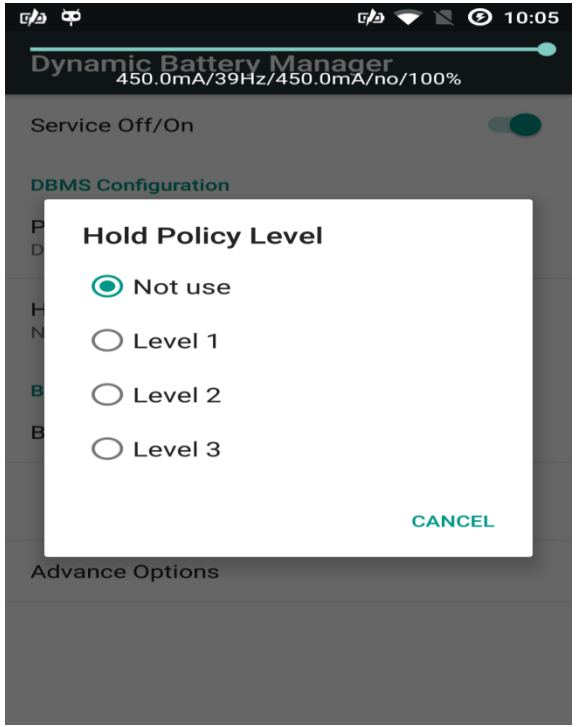

IDBMSLowService
+getDBMSDipCtrlValue(int)
+getEstimatedFps()
+getVsyncRate()
+requestIconProcessing(const char*, const char*)
+setBatteryState(int)
+setDBMSDipCtrlValue(int, int)
+setEstimatedFps(int)
+setVsyncRate(int)
+IDBMSLowService()
+derived
DBMSLowService
-mVsyncHitRatio
-mFps
-original_kcals[9]
-previous_kcals[9]
-cpu_frequency[100]
-cpu_freq_count
-isFlappedToOriginalColor
-mPrevSat
-drawIconBorder(unsigned char*, int, int, int)
-encodeOneStep(const char*, unsigned char*, unsigned int, unsigned int)
-imageProcessingInternal(const char*, const char*)
-process1(unsigned char*, char*, int, int)
-process2(unsigned char*, unsigned char*, int, int)
-process3(unsigned char*, unsigned char*, int, int)
-setDBMSDipCtrlValueInternal(int, int, bool)
-setGaussianFilter(unsigned char*, int, int)
-sobelEdgeColored(unsigned char*, unsigned char*, int, int)
-sobelEdgeDetector(unsigned char*, int, int)
+DBMSLowService()
+instantiate()
ISurfaceComposer
+notifyToInitiateLowService()
+setVsyncRate(int)
com_android_server_DBMSLowService
+notifyInitLowService()
+setVsyncRate(int)
+getEstimatedFps()
+setDBMSDipCtrlValue(int, int)
+getDBMSDipCtrlValue(int)
+setBatteryState(int)
+requestIconProcessing(String, String)
SurfaceFlinger
-vsyncrate
-mDBMSLowService
+getVsyncRate()
+getDBMSLowService()
+Bidirected Association
EventThread
-isGeneratedBySF
-current_t
-previous_t
-fps
-vsynchitcount
-mFlinger
+EventThread(const SurfaceFlinger&, const VsyncSource&, bool)
+onVsyncEvent(nsecs_t)


DBMSSettings
-bR
-mServiceStatus
-mSysService
-mBatteryService
-mAdvancedOptions
-mPolicyList
-mHoldModeList
-mKeeperStatus
-mKeeperSeekerBar
-mCallbackFromPreference
-notifyDBMSPolicyChanged()
-updateUIwithRecentInformation()
DBMSFloatService
-mWM
-mInflater
-mInfo
-mBatteryLevel
-mBatteryLevelChanged
-mLinearLayout
-mGlobalParams
-mReceiverFromOthers
-mLastBatteryLevel
-isShow
-mSysService
-mBatteryService
-mDBMSNotiMgr
-mContext
-mWatcherInfo
-mH
-readCurrentBatteryStatus()
-init()
-makeView()
+setNotification(boolean)
+onStartCommand(Intent, int, int)
+Bidirected Association
WatcherThread
-activatedFlag
+requestStop()
+requestStart()
+isActivated()
+run()
UI Relative Classes
DBMSDebugPreference
-mDebugList
-mShowData
-mVsyncRate
-mSysService
-mGrayscale
-mInvert
-mRed
-mGreen
-mBlue
-mSaturation
-mValue
-mContrast
-mHue
-mRevert
-mCallbackFromPreference
-sendCurrentProgressToFloatService()
-setData(String, String)
-getData(String, String)
-setDBMSDipData(int, int)
-getDBMSDipData(int)
-initUI()
VsyncRateSeekBar
-mCallback
-mKey
-updateCurrentRate()
+invalidate()
+setOnSeekBarChangedListener(OnSeekBarChangedListener)
+progressToLowFPSCostant(int)
+onProgressChanged(SeekBar, int, boolean)
+onStopTrackingTouch(SeekBar)
DBMSKcalCtrlSeekBar
-mCallback
-mKey
+setOnseekBarChangedListener(onSeekBarChangedListener)
+onProgressChanged(SeekBar, int, boolean)
+onStopTrackingTouch(SeekBar)
Debug Only


Launcher
-service
-imageIconMap
-batteryStateIconChangeFlag
-mDBServiceOnOff
+getDBServiceOnoff()
+connectDBService()
+iconImageMapSetting()
+getImageIconMap()
+getBatteryStateIconChangeFlag()
+onReceive(Context, intent)


10:04
Dynamic Battery Manager
450.0mA/30Hz/450.0mA/no/100%
Service Off/On
DBMS Configuration
Policy
Default
Hold Policy Level
Not use
Battery Keeper
Battery Keeper Off/On
20%
Advance Options


10:04
Dynamic Battery Manager
450.0mA/46Hz/450.0mA/no/100%
Service Off/On
DBMS Configuration
Policy
Default
PowerSave
Hybrid
CANCEL
Advance Options


10:05
Dynamic Battery Manager
450.0mA/39Hz/450.0mA/no/100%
Service Off/On
DBMS Configuration
Hold Policy Level
Not use
Level 1
Level 2
Level 3
CANCEL
Advance Options