

# **Project Minikin**

Raph Levien  
Android Frameworks  
Jan 2013

# Progress to date

## Complex text rendering

- JB: GSUB tables, RTL scripts
- JB MR1: GPOS (mark positioning)
- JB MR2: Harfbuzz NG
  - High fidelity script shaping

## Styling

- Fake bolding
- Language based font selection (JP)

# Hacks and limitations

- No fake italics
- Only two weights (Roboto has 6)
  - Workaround: separate families
- 64k total glyph limit (baseGlyphCount)
- Only one fallback list
  - Typeface.SERIF + "여보" → 여보
- Crude text measurement
  - Important for cursor positioning, selection



# Project Minikin

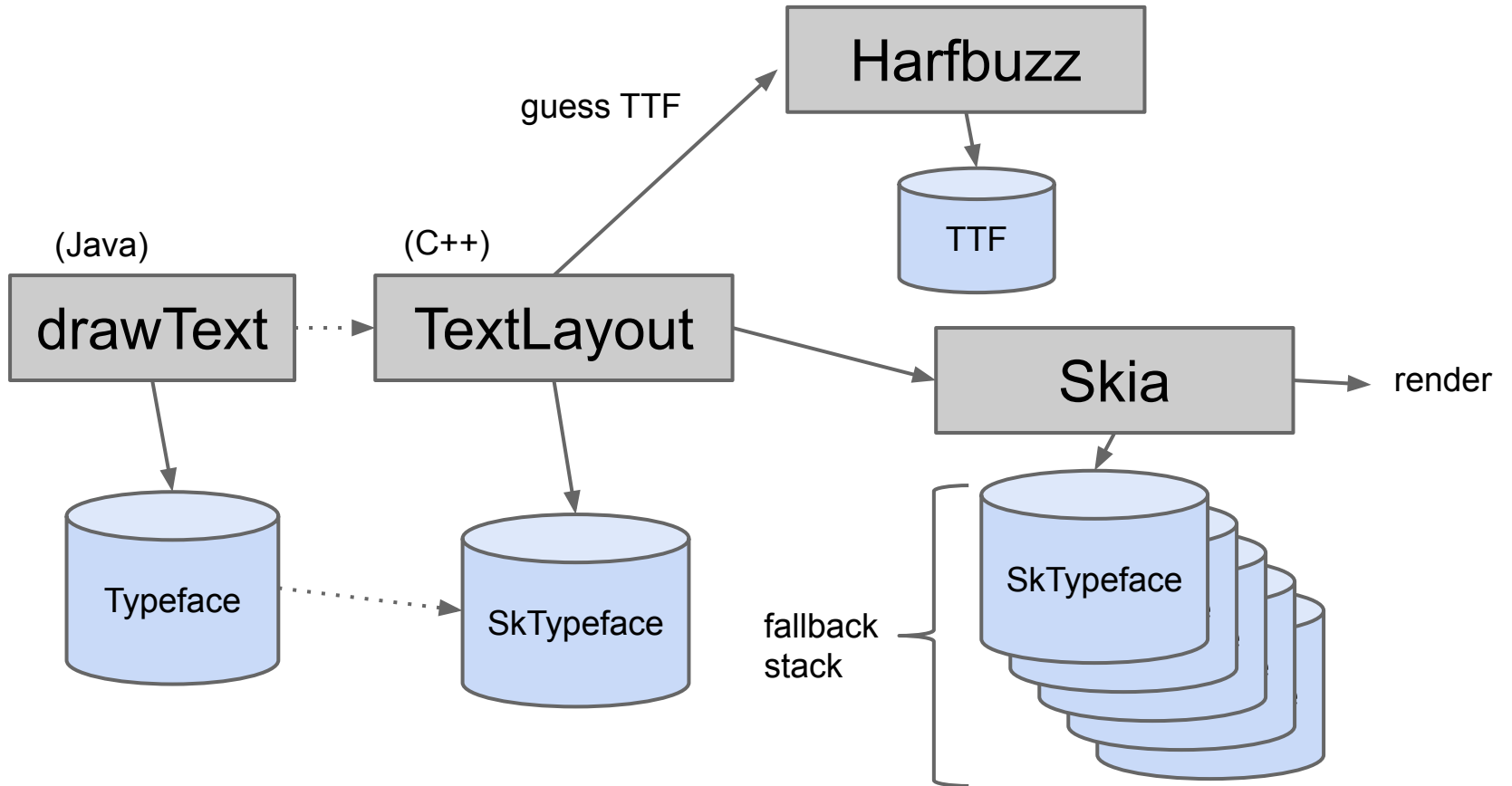
"Minikin" is old-style name for 3pt type

- Small
- C++
- Minimal dependencies
  - FreeType
  - Harfbuzz NG
  - ICU
- High quality

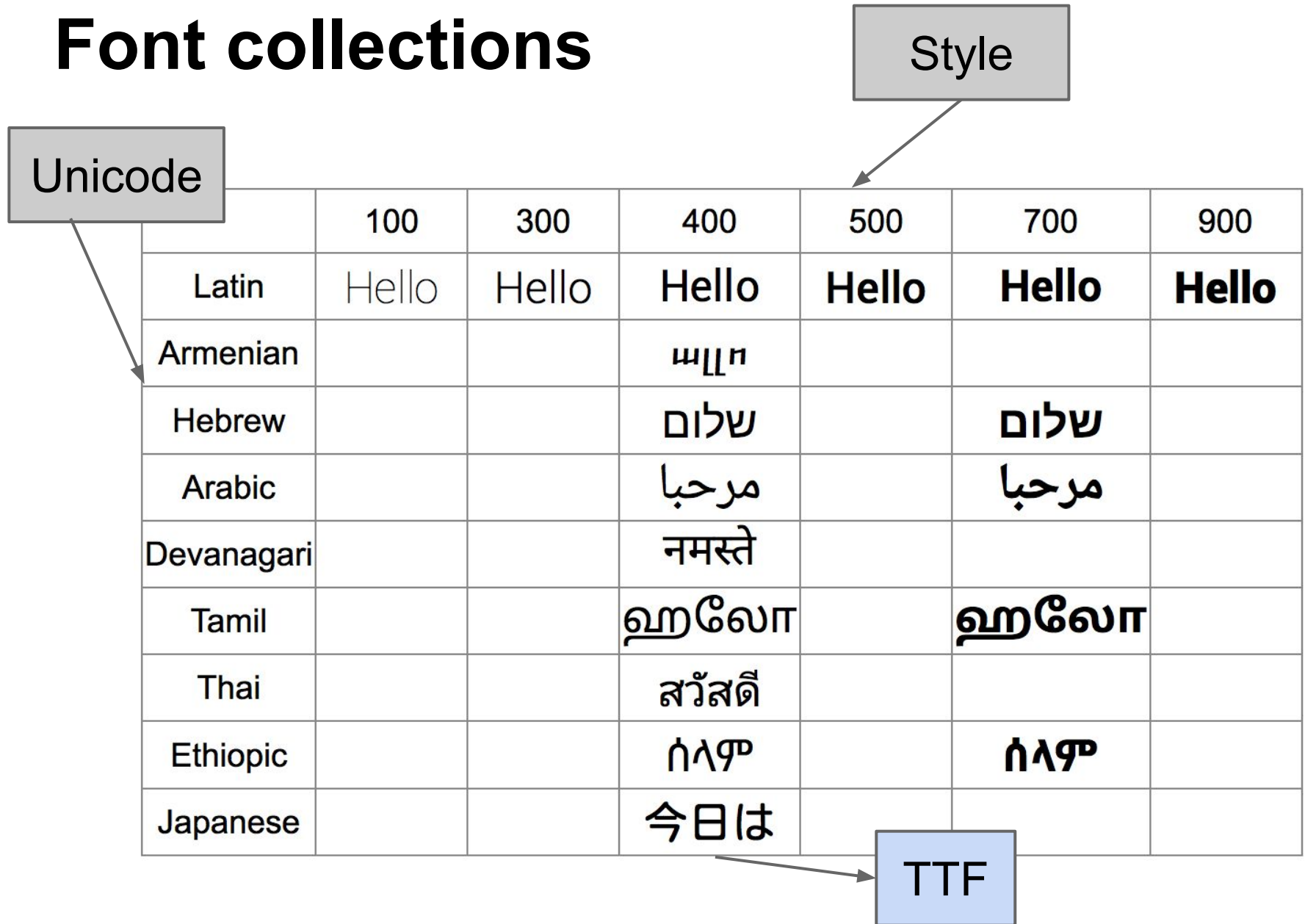
# Minikin goals

- More accurate text measurement
  - cursor ↔ (x,y)
- Richer styles
  - 100-900 weight (instead of bool bold)
  - Letterspacing
  - OpenType goodies (small caps, etc)
- Justification
  - Looks better & is more space-efficient
  - Stretch: paragraph-optimization too
- Richer layout
  - Indentation, margins, etc.
- Low impedance mismatch with CSS3

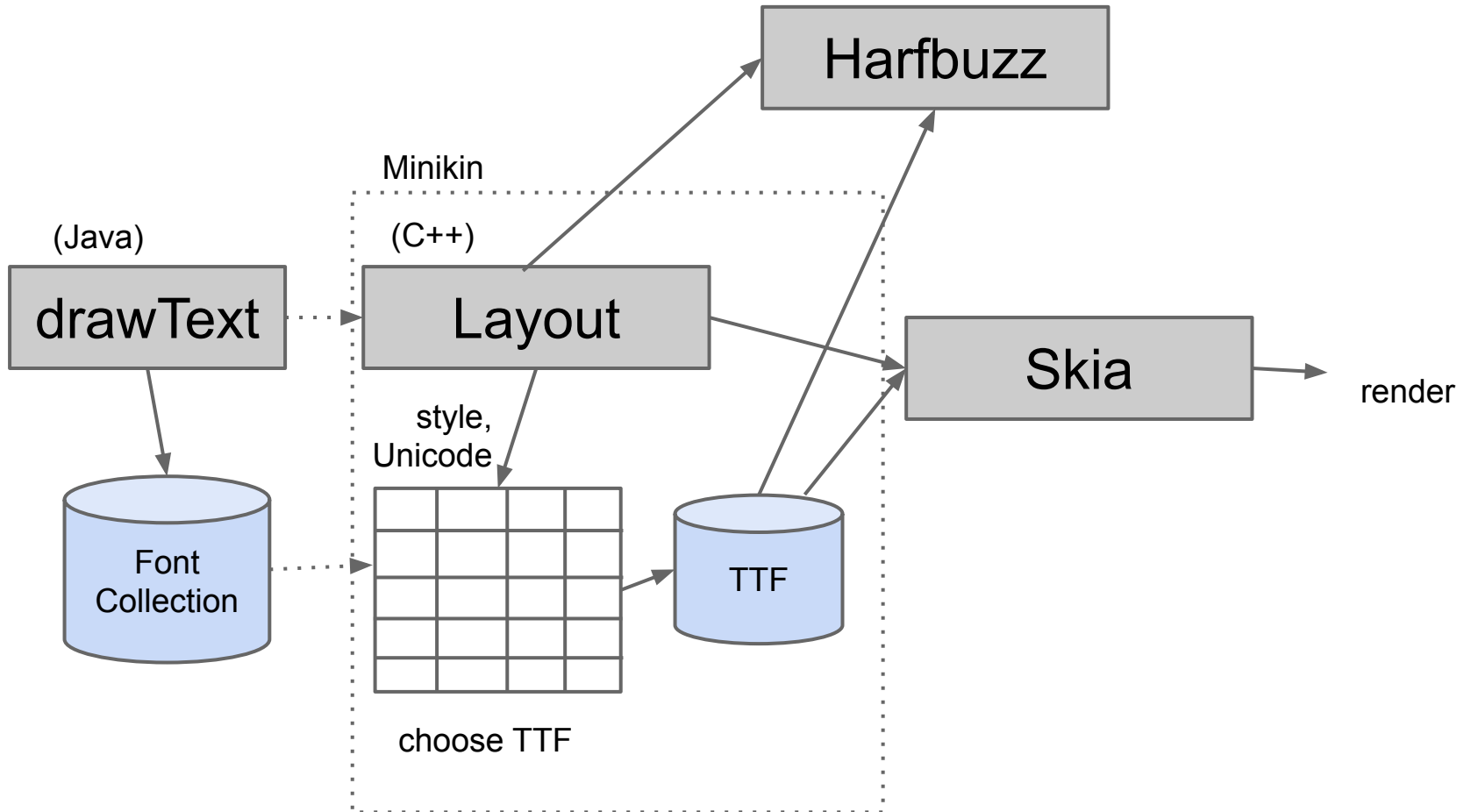
# Font rendering pipeline today



# Font collections



# Font rendering pipeline: Minikin





# Minikin deployment

## GEM

- Native code
- Use Romain's scene graph for rendering

## Key Lime Pie

- Needs Java APIs (some new)

## Compatibility lib

- Build as NDK (but export Java APIs only)
- Docs would be big customer

# Open questions

Retrofit existing stack to use new renderer, or build separate v2?

- Possible RichTextEdit widget

Put in compatibility library?

- Probably after it's had time to bake.

# Line height "abnormal"

	100	300	400	500	700	900
Latin	Hello	Hello	Hello	<b>Hello</b>	<b>Hello</b>	<b>Hello</b>
Armenian			ԿԼԼԻ			
Hebrew			הלוא		<b>הלוא</b>	
Arabic			مرحبا		<b>مرحبا</b>	
Devanagari			नमस्ते			
Tamil			ஹலோ		<b>ஹலோ</b>	
Thai			สวัสดี			
Ethiopic			ሰላም		<b>ሰላም</b>	
Japanese			今日は			