

Axya Slide Design System

Brand Guide for LLM Reproduction

This document defines the complete visual system used across Axya's presentation decks. It is intended to be fed to an LLM such as Claude so that any new slide can be produced in a style consistent with the existing deck.

1. Stack & Environment

Slides are built programmatically using pptxgenjs (Node.js).

None

```
npm install pptxgenjs
```

```
node build_deck.js
```

Slide dimensions: LAYOUT_16x9, 10" × 5.625" (914400 × 5143500 EMU).

After every build, run rezip.py then preview with LibreOffice and pdftoppm. LibreOffice substitutes system fonts; QA previews may look slightly different from the actual file opened in PowerPoint.

2. Color System

Every color is a 6-character hex string (no #).

JavaScript

```
const BLUE    = "2682FD"; // Axya brand blue: CTAs, highlights,  
borders, circles, ALL icons
```

```
const BBLUE   = "EBF3FF"; // Very light blue: active card  
backgrounds, tinted panels
```

```
const GREY    = "F4F4F8"; // Near-white grey: default card  
background
```

```
const WHITE = "FFFFFF"; // Pure white: slide background, chip
fills

const DARK = "12123A"; // Deep navy: all headings and large
titles

const MID = "4A4A6A"; // Medium purple-grey: body text, bullet
items

const MUTED = "9090B0"; // Light grey-purple: captions, labels,
dividers

const GREEN = "22C55E"; // Success green: checkmarks, completed
stages

const ORANGE = "E07030"; // Amber-orange: ASK labels, warning
states


// Extended palette used in specific slides:

const WARN_BG = "FFF7ED"; // Very light orange: warning card
background

const WARN_BD = "FED7AA"; // Light amber: warning card border

const SUCC_BG = "F0FBF4"; // Very light green: success card
background

const SUCC_BD = "86EFAC"; // Light green: success card border

const PURPLE = "5B21B6"; // Deep purple: secondary accent
(AI/Operations columns)

const PURPLE_BG= "F5F0FF"; // Very light purple: purple card
background

const PURPLE_BD= "C4B5FD"; // Light purple: purple card border

const RED = "DC2626"; // Warning red: supply chain cost icon

const RED_BG = "FEF2F2"; // Very light red: red card background
```

```
const RED_BD    = "FECACA"; // Light red: red card border
```

Background rule: All slide backgrounds are *WHITE*. Never use *GREY* or *BBLUE* as a slide background; use them only for card fills.

Accent rule: *BLUE* is the only accent color used for interactive/active states, borders on highlighted cards, circles, and CTA buttons. Use sparingly.

3. Typography

Single font throughout: "Google Sans Flex". No exceptions, anywhere in the file.

- No fallbacks. The font name must be passed exactly as written above. Never use "Calibri", "Arial", or any other font.
- Every `addText` call must carry `fontFace: "Google Sans Flex"`. The safest pattern is to route all text through a single helper that injects it (see Section 12).
- Set the `pptxgenjs` theme so the file's default fonts are also Google Sans Flex:

JavaScript

```
pres.theme = { headFontFace: "Google Sans Flex", bodyFontFace:
"Google Sans Flex" };
```

- **Theme XML scrub (required):** even with `pres.theme` set, `pptxgenjs` writes Arial and per-script fallback fonts (Segoe UI, Times New Roman, CJK faces, etc.) into `ppt/theme/theme1.xml` and `ppt/slideMasters/slideMaster1.xml`. After building, unpack the file and replace every `typeface="..."` value that is not *Google Sans Flex*, not a theme reference (`+mj-lt`, `+mn-lt`, etc.), and not empty, with `typeface="Google Sans Flex"`. Then repack and run `rezip.py`. Verify with:

Shell

```
grep -rhoP 'typeface="[^\"]*"' unpacked/ppt --include="*.xml" |
sort | uniq -c
```

The only surviving values must be *Google Sans Flex*, `+mj-*` / `+mn-*` references, and empty strings.

Font Size Scale

<i>Use</i>	<i>Size</i>	<i>Weight</i>	<i>Color</i>
<i>Section kicker label (eyebrow)</i>	9–9.5pt	<i>bold</i>	<i>BLUE</i>
<i>Slide title (main)</i>	28–32pt	<i>normal (400)</i>	<i>DARK</i>
<i>Slide subtitle</i>	12–12.5pt	<i>normal</i>	<i>MID</i>
<i>Card title (large)</i>	17–22pt	<i>bold</i>	<i>DARK</i>
<i>Card title (narrow cards, ≤2.2" wide)</i>	12–13pt	<i>bold</i>	<i>DARK</i>
<i>Section label inside card (pill)</i>	8–9pt	<i>bold</i>	<i>BLUE or WHITE</i>
<i>Body / bullet items</i>	10.5–12pt	<i>normal</i>	<i>MID</i>
<i>Captions / meta labels</i>	9–10pt	<i>normal</i>	<i>MUTED</i>
<i>Big stat numbers</i>	32–44pt	<i>normal (400)</i>	<i>DARK</i>
<i>Stat sub-labels</i>	10–11pt	<i>normal</i>	<i>MID</i>

Critical rules

- Content-slide titles are always weight 400 (not bold). Never use *bold:true* on a content slide title. Exception: cover and divider slides (Section 9.6) use bold titles.
- Card titles use *bold:true*.

- **No letter-spacing anywhere.** Never use `charSpacing` on any text element. Kicker labels, section labels, and pills are bold and colored but tracked normally. (This replaces the old `charSpacing 2.5` kicker rule.)
- Never use `italic:true` on visible slide content. Italic is reserved for internal speaker-note quotes that have been explicitly requested.
- No em dashes anywhere. Use colons, commas, or periods instead.
- No sales questions, coaching notes, or speaker-only lines on slides.

4. Icon System

All icons are flat glyph PNGs rendered in brand blue.

- **Color: every icon is exactly BLUE #2682FD.** If a source icon is any other color, recolor it before use: set the RGB of every non-transparent pixel to 2682FD while preserving the alpha channel. Keep near-white interior details (checkmarks, calendar squares: pixels with $\min(R,G,B) > 235$) white.
- **Watch for all-white glyph variants.** Some icon exports are white-on-transparent and disappear on light cards. If an icon has zero colored pixels, recolor all opaque pixels to 2682FD.
- **Sizes (keep icons small):**
 - 0.24" square: default for icon + bold-title cards (3-column grids, 2x2 proof grids)
 - 0.26" square: compact capability rows
 - 0.28" square: centered channel chips, inline strip icons (e.g. deal-protection shield)
 - 0.32" square: hero/kickoff cards (maximum; never larger)
- **Placement:** top-left of the card at `x: cx + 0.2`, optically centered against the adjacent title line (nudge y down 0.02–0.05" relative to the title y). Card title starts at `x: cx + 0.6` regardless of icon size, so the gap stays consistent.
- Never place icons in colored circles; the glyph sits directly on the card fill.
- Recolor helper (Python/Pillow):

Python

```
from PIL import Image

BLUE = (0x26, 0x82, 0xFD)

im = Image.open(path).convert("RGBA"); px = im.load()

for y in range(im.height):
    for x in range(im.width):
        r, g, b, a = px[x, y]
```

```
        if a == 0: continue

        if min(r, g, b) > 235: continue    # keep white interior
details

        px[x, y] = (*BLUE, a)

im.save(path)
```

5. Layout Grid

None

Slide width: 10"

Slide height: 5.625"

Left margin: SX = 0.45"

Right edge: RX = 9.55"

Content width: W = 9.1" (RX - SX)

Safe top: 0.28"

Safe bottom: 5.32"

Vertical zones for slide elements:

None

0.28" – Section kicker label (9.5pt, BLUE, bold, h=0.26")

0.56" – Main title (30–32pt, DARK, h=0.52")

1.10" – Subtitle (12pt, MID, h=0.32")

Cards with subtitle above: start at $y = 1.56''$ max (card bottom $\leq 5.32''$)

Cards with title only above: start at $y = 1.22''$ max (card bottom $\leq 5.32''$)

Max card height with subtitle: $5.32 - 1.56 = 3.76''$

Max card height with title only: $5.32 - 1.22 = 4.10''$

Always verify: $card_y + card_h \leq 5.32''$. Overflow causes clipping in PowerPoint.

Always verify: last text element inside card ends before $card_y + card_h - 0.08''$.

6. Shadow System

Every card and elevated shape uses the same outer drop shadow:

JavaScript

```
const sh = () => ({ type:"outer", color:"000000", blur:8,
offset:2, angle:45, opacity:0.06 });
```

```
const sh2 = () => ({ type:"outer", color:"000000", blur:14,
offset:3, angle:45, opacity:0.09 });
```

`sh()` is the default. `sh2()` is used only for the highlighted Axya box in the ERP diagram.

Rule: Always call `sh()` as a function to return a fresh object. `pptxgenjs` mutates shadow objects, so reusing the same reference causes errors.

7. Shape Primitives

7.1 Cards

Standard rounded rectangle:

JavaScript

```
s.addShape(pres.shapes.ROUNDED_RECTANGLE, {  
  
  x, y, w, h,  
  
  fill:      { color: GREY },  
  
  line:      { color: "E4E4F0", width: 0.5 },  
  
  rectRadius: 0.18,  
  
  shadow:    sh()  
  
});
```

Radius values:

- 0.18: standard card
- 0.12: compact card or chip row
- 0.10: small chip (method row, ERP name chip)
- 0.08: tiny pill (kicker badge inside card)
- 0.05: plus-icon button

Active / highlighted card (e.g. F1 in the framework): `fill: { color: BBLUE }, line: { color: BLUE, width: 1.5 }, rectRadius: 0.18`

Warning card: `fill: { color: WARN_BG }, line: { color: WARN_BD, width: 0.75 }` *Success card:* `fill: { color: SUCC_BG }, line: { color: SUCC_BD, width: 0.75 }`

7.2 Filled Circles

Used for numbered steps, F-labels, checkmarks:

JavaScript

```
s.addShape(pres.shapes.OVAL, {  
  
  x, y, w: r, h: r,  
  
  fill: { color: BLUE },  
  
  line: { color: BLUE }  
  
});
```

```
s.addText(label, {
  x, y, w: r, h: r,
  fontSize: 12, fontFace: "Google Sans Flex",
  bold: true, color: WHITE,
  align: "center", valign: "middle", margin: 0
});
```

Standard circle sizes: $r = 0.5$ " (large), $r = 0.44$ " (medium), $r = 0.28$ " (small bullet). Never use `pres.shapes.OVAL` with `endArrowType`; use `LINE` for arrows.

7.3 Horizontal Divider Lines

JavaScript

```
s.addShape(pres.shapes.LINE, {
  x, y, w, h: 0,
  line: { color: "D8D8EC", width: 0.5 }
});
```

Use "D8D8EC" for standard dividers. Use "C8D8F8" for dividers inside BBLUE cards. Use the card's own border color for dividers inside colored cards (e.g. SUCC_BD, WARN_BD).

7.4 Arrows

Horizontal arrow (connector between cards):

JavaScript

```
s.addShape(pres.shapes.LINE, {
  x: x1, y, w: x2 - x1, h: 0,
  line: { color: "C0C0D0", width: 1, endArrowType: "arrow" }
});
```

Vertical arrow (flow diagrams):

JavaScript

```
s.addShape(pres.shapes.LINE, {  
  x, y: y1, w: 0, h: y2 - y1,  
  line: { color: BLUE, width: 1.5, endArrowType: "arrow" }  
});
```

Rule: Always use *LINE* shapes for arrows, never Unicode arrow characters (→ ►). Unicode characters render in wrong colors under LibreOffice.

7.5 Plus-Icon Bullets

Used in the NBM deck for feature/benefit lists:

JavaScript

```
s.addShape(pres.shapes.ROUNDED_RECTANGLE, {  
  x, y, w: 0.28, h: 0.28,  
  fill: { color: BLUE }, line: { color: BLUE }, rectRadius: 0.05  
});  
  
s.addText("+", {  
  x, y, w: 0.28, h: 0.28,  
  fontSize: 13, fontFace: "Google Sans Flex",  
  bold: true, color: WHITE,  
  align: "center", valign: "middle", margin: 0  
});
```

7.6 Accent Bar (thin horizontal stripe above stats)

JavaScript

```
s.addShape(pres.shapes.ROUNDED_RECTANGLE, {  
  x: cx + 0.22, y: cy + 0.24, w: 0.32, h: 0.06,
```

```
    fill: { color: BLUE }, line: { color: BLUE }, rectRadius: 0.03
  });
```

Used on stat cards: tiny blue bar above the large number.

7.7 Call-Mapping Pills (bottom of F-framework cards)

JavaScript

```
s.addShape(pres.shapes.ROUNDED_RECTANGLE, {
  x: cx + 0.2, y: cy + ch - 0.36, w: cw - 0.4, h: 0.26,
  fill: { color: isActive ? BLUE : "E0E0EE" },
  line: { color: isActive ? BLUE : "E0E0EE" },
  rectRadius: 0.08
});

s.addText(callLabel, {
  x: cx + 0.2, y: cy + ch - 0.36, w: cw - 0.4, h: 0.26,
  fontSize: 7.5, fontFace: "Google Sans Flex",
  bold: true, color: isActive ? WHITE : MUTED,
  align: "center", valign: "middle", margin: 0
});
```

7.8 Blue CTA Button

JavaScript

```
s.addShape(pres.shapes.ROUNDED_RECTANGLE, {
  x, y, w, h,
  fill: { color: BLUE }, line: { color: BLUE }, rectRadius: 0.12
});
```

```
s.addText(label, {
  x, y, w, h,
  fontSize: 13, fontFace: "Google Sans Flex",
  bold: true, color: WHITE,
  align: "center", valign: "middle", margin: 0
});
```

Pill sizing rule: pills never wrap. Size the pill so the label fits on one line (roughly 0.09" per character at 7.5–8pt bold); if it doesn't fit, widen the pill or shorten the label, never shrink below 7.5pt.

7.9 Stat Pill (inside card, light blue)

JavaScript

```
s.addShape(pres.shapes.ROUNDED_RECTANGLE, {
  x: cx + 0.22, y: ..., w: cw - 0.44, h: 0.44,
  fill: { color: BBLUE }, line: { color: "C8D8F8", width: 0.5 },
  rectRadius: 0.10
});

s.addText(statText, {
  ..., fontSize: 11, fontFace: "Google Sans Flex",
  bold: true, color: BLUE, align: "center", valign: "middle",
  margin: 0
});
```

7.10 "SHOW IF ASKED" Badge

For appendix/optional slides:

JavaScript

```
s.addShape(pres.shapes.ROUNDED_RECTANGLE, {
```

```
    x: SX, y: 0.28, w: 2.6, h: 0.3,

    fill: { color: "FEF3C7" }, line: { color: "FCD34D", width: 0.5
}, rectRadius: 0.08

});

s.addText("SHOW IF ASKED", {

    x: SX, y: 0.28, w: 2.6, h: 0.3,

    fontSize: 8, fontFace: "Google Sans Flex",

    bold: true, color: "92400E",

    align: "center", valign: "middle", margin: 0

});
```

8. Text Rules

Margin

Always `margin: 0` on every `addText` call. The `pptxgenjs` default margin adds unwanted padding.

Alignment

- Slide titles: left-aligned
- Kicker labels: left-aligned
- Inside full-width circles (icon buttons): `align:"center", valign:"middle"`
- Inside pills and CTA buttons: `align:"center", valign:"middle"`
- All other body text: left-aligned

Bold

- `bold: true`: card titles, kicker labels, section labels, stat sub-labels, CTA text
- `bold: false` (omit): slide titles (important: never bold a slide title), body text, stat numbers

Letter-spacing

None. Never pass `charSpacing` on any element.

Line breaks in titles

Slide titles use a single `addText` with enough height to allow wrapping, not two separate calls. If a title must be on exactly one line, reduce font size until it fits rather than splitting.

9. Slide Structure Patterns

9.0 Slide Furniture (every content slide)

- Small Axya logo top-right on every slide except the cover: fit inside `x: RX - 0.78`, `y: 0.26`, `w: 0.78`, `h: 0.34`. Add it in the shared footer helper so it can never be forgotten.
- Page number bottom-right (8.5pt, MUTED, right-aligned at `x: RX - 0.6`, `y: 5.4`). No page number on the cover.
- Any element in the top-right corner (badges, pills) must start at `y ≥ 0.66` so it clears the corner logo.

9.1 Standard Slide Header (with section label)

JavaScript

```
// Kicker label

s.addText("SECTION NAME", {

  x: SX, y: 0.28, w: W, h: 0.26,

  fontSize: 9.5, fontFace: "Google Sans Flex",

  bold: true, color: BLUE, margin: 0

});

// Main title

s.addText("Slide Title", {

  x: SX, y: 0.56, w: W, h: 0.52,

  fontSize: 30, fontFace: "Google Sans Flex", color: DARK, margin:

0
```

```

});

// Optional subtitle

s.addText("One sentence that frames the slide.", {

  x: SX, y: 1.12, w: W, h: 0.32,

  fontSize: 12.5, fontFace: "Google Sans Flex", color: MID,
margin: 0

});

// Cards start at y = 1.56 when subtitle is present

// Cards start at y = 1.22 when no subtitle

```

9.2 Three-Column Equal Card Grid

```

None

Total width = W = 9.1"

cw * 3 + gap * 2 = 9.1

Standard: cw = 2.84, gap = 0.29

const cw = 2.84, gap = 0.29, cy = 1.56, ch = 3.62;

// col 0: cx = SX                = 0.45

// col 1: cx = SX + cw + gap    = 3.58

// col 2: cx = SX + 2*(cw+gap)= 6.71

```

9.3 Four-Column Equal Card Grid (narrow cards)

```

None

cw * 4 + gap * 3 = 9.1

Standard: cw = 2.12, gap = 0.18 (for flow/friction/outcome cards)

```

```
const cw = 2.12, gap = 0.18, cy = 1.22, ch = 3.88;
```

For the 4-F framework (wider cards with more content):

None

```
cw = 2.12, arrW = 0.22, gap = 0.04    (arrow width between cards)
```

```
colX(i) = SX + i * (cw + arrW + gap)
```

9.4 Two-Column Split (left + right)

JavaScript

```
// Standard 40/60 split:
```

```
const lw = 4.4, gap = 0.22;
```

```
const rw = W - lw - gap; // = 4.48
```

```
const lx = SX, rx = SX + lw + gap;
```

```
// Standard 50/50 split:
```

```
const half = (W - 0.18) / 2; // ≈ 4.46
```

```
const lx = SX, rx = SX + half + 0.18;
```

9.5 Full-Width Single Card

JavaScript

```
card(s, SX, cardY, W, cardH);
```

```
// Internal 2-column grid inside card:
```

```
const cellW = (W - 0.52 - 0.12) / 2; // ≈ 4.23 each
```

```
const gridX = [SX + 0.26, SX + 0.26 + cellW + 0.12];
```

```
const gridY = [cardY + 0.25, cardY + 0.25 + cellH + 0.12];
```

9.6 Title & Divider Slides (cover, section breaks, appendix, closing)

Cover and divider slides follow one fixed format, based on the Axya NBM template:

- Full-slide background art: the diagonal Axya document-card collage image, placed full-bleed at $x:0$, $y:0$, $w:10$, $h:5.625$. The asset is a 16:9 image with a clean white left half; all text sits on that white area. Reuse the extracted asset (`title_art.png`) rather than redrawing it.
- Axya logo top-left: $x: SX$, $y: 0.42$, fit inside 1.3×0.55 .
- Kicker (deck or section name, all caps): $x: SX$, $fontSize: 13$, **bold**, **BLUE**. On the cover it sits at $y \approx 1.66$; on dividers it can sit lower ($y \approx 2.1$) so the block is vertically centered.
- Title: **bold**, 30pt, **DARK**, $w: 4.9$, up to 3 lines. This is the one place bold slide titles are allowed. Break the title manually so no line exceeds ~20 characters at 30pt bold in the 4.9" column.
- Below the title, one small MID line: the date on the cover (e.g. "July 2026", 12pt), or a one-sentence description on dividers.
- Cover only: customer-logo placeholder bottom-left, rendered as the text "LOGO" in 30pt bold **ORANGE**, to be replaced with the prospect's logo per deck.
- No footer URL, no page number on the cover; no pills, chips, or cards on these slides.

Example (cover):

JavaScript

```
s.addImage({ path: A("title_art"), x: 0, y: 0, w: 10, h: 5.625 });

fitImg(s, "axya_logo", SX, 0.42, 1.3, 0.55);

T(s, "THE AXYA PARTNER PROGRAM", { x: SX, y: 1.66, w: 4.9, h: 0.3,
  fontSize: 13, bold: true, color: BLUE });

T(s, "Turn Every ERP\nEngagement into a\nProcurement Win.", { x:
  SX, y: 2.04, w: 4.9, h: 1.85, fontSize: 30, bold: true, color:
  DARK });

T(s, "July 2026", { x: SX, y: 4.06, w: 3.0, h: 0.28, fontSize: 12,
  color: MID });

T(s, "LOGO", { x: SX, y: 4.62, w: 2.5, h: 0.55, fontSize: 30,
  bold: true, color: ORANGE });
```

10. Component Recipes

10.1 Stat Card

JavaScript

```
card(s, cx, cy, cw, ch); // ch ≈ 3.6"

// Accent bar
s.addShape(pres.shapes.ROUNDED_RECTANGLE, {
  x: cx + 0.22, y: cy + 0.24, w: 0.32, h: 0.06,
  fill: { color: BLUE }, line: { color: BLUE }, rectRadius: 0.03
});

// Big stat number
s.addText(stat, {
  x: cx + 0.2, y: cy + 0.42, w: cw - 0.4, h: 0.82,
  fontSize: 44, fontFace: "Google Sans Flex", color: DARK, margin:
0
});

// Bold label
s.addText(label, {
  x: cx + 0.2, y: cy + 1.28, w: cw - 0.4, h: 0.36,
  fontSize: 12, fontFace: "Google Sans Flex", bold: true, color:
DARK, margin: 0
});
```

```

// Divider

hline(s, cx + 0.2, cy + 1.68, cw - 0.4);

// Sub-description

s.addText(desc, {
    x: cx + 0.2, y: cy + 1.8, w: cw - 0.4, h: 1.58,
    fontSize: 10.5, fontFace: "Google Sans Flex", color: MID,
    margin: 0
});

```

10.2 Numbered Column Card

```

JavaScript

card(s, cx, cy, cw, ch);

// Numbered circle (top-left)

circ(s, cx + 0.2, cy + 0.2, 0.44, BLUE);

circT(s, cx + 0.2, cy + 0.2, 0.44, String(num), 13);

// Card title (bold, ~13pt for narrow cards)

s.addText(title, {
    x: cx + 0.18, y: cy + 0.74, w: cw - 0.36, h: 0.42,
    fontSize: 13, fontFace: "Google Sans Flex", bold: true, color:
    DARK, margin: 0
});

```

```

// Divider

hline(s, cx + 0.18, cy + 1.2, cw - 0.36);

// Section label (e.g. "PAIN POINTS")
s.addText("PAIN POINTS", {
    x: cx + 0.18, y: cy + 1.28, w: cw - 0.36, h: 0.22,
    fontSize: 7.5, fontFace: "Google Sans Flex",
    bold: true, color: BLUE, margin: 0
});

// Items
items.forEach((item, j) => {
    s.addText("· " + item, {
        x: cx + 0.18, y: cy + 1.52 + j * 0.34, w: cw - 0.36, h: 0.32,
        fontSize: 9.5, fontFace: "Google Sans Flex", color: MID,
        margin: 0
    });
});

// "ASK" label (for discovery slides)
s.addText("ASK", {
    x: cx + 0.18, y: qy, w: cw - 0.36, h: 0.22,
    fontSize: 7.5, fontFace: "Google Sans Flex",

```

```
    bold: true, color: ORANGE, margin: 0
  });
```

If bullet items can wrap to two lines, use a 0.5" row pitch instead of 0.34".

10.3 Icon Card (icon + bold title + description)

Standard cell for proof grids, expectations, and commitment slides:

JavaScript

```
card(s, cx, cy, cw, ch, GREY, "E4E4F0", 0.14);

fitImg(s, iconName, cx + 0.2, cy + 0.23, 0.24, 0.24); //
brand-blue icon, small

s.addText(title, {
  x: cx + 0.6, y: cy + 0.16, w: cw - 0.8, h: 0.5,
  fontSize: 12, fontFace: "Google Sans Flex", bold: true, color:
  DARK, margin: 0
});

s.addText(desc, {
  x: cx + 0.2, y: cy + 0.68, w: cw - 0.4, h: ch - 0.8,
  fontSize: 9.5, fontFace: "Google Sans Flex", color: MID, margin:
  0
});
```

Give the title *h: 0.5* so a two-line title never collides with the description.

10.4 Four-F Journey Cards (F1 → F2 → F3 → F4)

F1 through F3 use standard GREY fill; F4 (Future Value) uses BBLUE fill with BLUE border (1.2pt) to visually distinguish the aspirational outcome.

JavaScript

```
const isFuture = i === 3;
```

```

s.addShape(pres.shapes.ROUNDED_RECTANGLE, {
    x: cx, y: cy, w: cw, h: ch,
    fill:      { color: isFuture ? BBLUE : GREY },
    line:      { color: isFuture ? BLUE : "E4E4F0", width: isFuture ?
1.2 : 0.5 },
    rectRadius: 0.18, shadow: sh()
});

// Between each F-card, draw a horizontal arrow:

harrow(s, cx + cw + gap, cx + cw + gap + arrW, cy + ch / 2,
"C0C0D0");

```

10.5 ERP → Axya → Supplier Diagram (vertical flow)

None

Layout: Left column (4.5" wide, centered at x=2.5") + Right column (x=5.5", w=3.95")

```

const dc = 2.5;          // diagram horizontal center
const bw = 3.6;          // ERP + Supplier box width
const bx = dc - bw/2;    // = 0.7
const aw = 4.4;          // Axya box width (wider)
const ax = dc - aw/2;    // = 0.3

// ERP box:      x: bx, y: 1.28, w: bw, h: 0.68, GREY fill,
rectRadius 0.14

// varrow down:  dc - 0.01, 1.96 → 2.14, BLUE

```

```
// Axya box:      x: ax, y: 2.16, w: aw, h: 2.06, BBLUE fill, BLUE
border 1.5pt, shadow sh2()

// 5 capability chips inside Axya box (white fill, "C8D8F8"
border)

// varrow down:  dc - 0.01, 4.22 → 4.38, BLUE

// Supplier box: x: bx, y: 4.40, w: bw, h: 0.68, GREY fill

// Return sync note (bottom strip): x: SX, y: 5.12, w: 4.6, h:
0.26, GREY
```

10.6 Horizontal Connector Flow (integration diagrams)

Four boxes with arrows between them (e.g. Infor ERP → Axya Sync → Internet → Axya Suite):

```
JavaScript
const bw = 1.9, bgap = 0.5, by = 3.66, bh = 1.28;

// per box: card(...); highlight the destination box with BBLUE
fill + BLUE 1.2pt border

// between boxes: harrow(s, bx + bw + 0.06, bx + bw + bgap - 0.06,
by + bh / 2, "C0C0D0");

// per box, identical layout: icon/logo at (bx+0.16, by+0.16,
0.282),

// bold title at y = by+0.5 (10.5pt, h 0.44 for two lines),
sub-label at y = by+0.96 (8.5pt MID)
```

Icon/logo consistency rule (all diagrams and card rows): either every node/box in a flow, grid, or row carries an icon or logo in the same position and size, or none of them do. Never decorate only one box (e.g. only the ERP vendor logo) while its siblings have plain text; mixed decoration breaks alignment. If a matching logo or icon is unavailable for even one box, drop them from the whole group.

10.7 Buying Journey / Timeline (horizontal numbered steps)

None

```
// 4 steps, arrow between each

const cw = 2.08, gap = 0.14, cy = 1.52, ch = 3.38;

// Active step: BBLUE fill, BLUE border 0.75pt

// Decision step: SUCC_BG fill, SUCC_BD border 0.75pt

// Other steps: GREY fill, "E4E4F0" border 0.75pt


// Numbered circle centered above card title:

circ(s, cx + cw/2 - 0.24, cy + 0.22, 0.48, stepColor);

circT(s, cx + cw/2 - 0.24, cy + 0.22, 0.48, stepNum, 13);
```

10.8 Implementation Timeline (dots above cards)

JavaScript

```
const cw = 2.12, gap = 0.18, cy = 2.15, ch = 2.9;

const dotR = 0.4, dotY = 1.52;


// Horizontal connecting line

s.addShape(pres.shapes.LINE, {

  x: SX + dotR/2, y: dotY + dotR/2, w: W - dotR, h: 0,

  line: { color: "D0D0E8", width: 1.5 }

});


// For each step:

const cardX = SX + i * (cw + gap);

const dotX = cardX + (cw - dotR) / 2; // DOT CENTERED OVER CARD
```

```

circ(s, dotX, dotY, dotR, BLUE);

circT(s, dotX, dotY, dotR, String(i + 1), 13);

card(s, cardX, cy, cw, ch);

```

10.9 Progress Tracker (closing slide, vertical stepper)

JavaScript

```

const stages = [

  { label: "Friction Discovery", done: true, sub: "Completed
today", col: GREEN },

  { label: "Functionality Demo", done: false, sub: "Up next",
col: MUTED },

  { label: "Business Case", done: false, sub: "Final call",
col: MUTED },

];

const stageH = 1.2, stageY0 = cy + 0.74;

stages.forEach((st, i) => {

  const sy = stageY0 + i * stageH;

  const circR = 0.6;

  circ(s, cx + 0.28, sy + 0.08, circR, st.done ? GREEN :
"E0E0E8");

  circT(s, cx + 0.28, sy + 0.08, circR, st.done ? "✓" : st.f,
st.done ? 20 : 13);

  s.addText(st.label, { x: cx + 1.02, y: sy + 0.12, ... bold:
true, color: st.done ? DARK : MUTED });

  if (!st.done) {

```

```

        // BBLUE calendar pill, BLUE text
    }

    if (i < stages.length - 1) {

        s.addShape(pres.shapes.LINE, {

            x: cx + 0.28 + circR/2, y: sy + 0.08 + circR, w: 0, h:
stageH - circR - 0.14,

            line: { color: "D8D8EC", width: 1.2 }

        });

    }

});

```

10.10 Consultative Discovery Card ("What We Know About [Customer]")

Left card (research):

- Logo placeholder box (BBLUE fill, "C8D8F8" border, 1.1" × 0.72")
- 6 label-value rows: bold label + value
- Muted instruction: "Replace placeholders with your pre-call research."

Right card (questions):

- 3-column inner grid (Procurement / Challenges / Urgency)
- Column headers in BLUE bold
- Questions as "· question text?" bullet items, 9pt

10.11 Term/Definition Rows

Full-width chip rows with a tag pill on the right:

```

JavaScript
const ry0 = 1.66, rh = 0.62, rgap = 0.11;

card(s, SX, ry, W, rh, GREY, "E4E4F0", 0.12);

// term:  x: SX + 0.26, w: 2.75, 11pt bold DARK, valign middle
// desc:  x: SX + 3.15, w: 4.38, 9.5pt MID, valign middle

```

```
//          (desc width must stop  $\geq 0.18$ " before the pill)

// pill:  x: RX - 1.36, w: 1.12, h: 0.28

//          TARGET = BLUE fill / WHITE text; DEFINITION = "E0E0EE"
fill / MUTED text
```

11. Color Usage by Slide Type

Slide purpose	Card background	Border	Circle / icon accent
Framework / F1–F4	GREY (F1–F3), BBLUE (F4/active)	"E4E4F0" (F1–F3), BLUE (F4)	BLUE
Discovery / pain points	GREY	"E4E4F0"	BLUE
Customer proof / outcomes	GREY (profile), SUCC_BG (results)	"E4E4F0", SUCC_BD	GREEN
ROI / hard savings	BBLUE	BLUE	BLUE
Productivity / operations	PURPLE_BG	PURPLE_BD	PURPLE
Strategic value	SUCC_BG	SUCC_BD	GREEN
Warning / cost / "without"	WARN_BG	WARN_BD	ORANGE

<i>Current state</i>	<i>GREY</i>	<i>"E4E4F0"</i>	<i>WARN_BD dots</i>
<i>Future state</i>	<i>BBLUE</i>	<i>BLUE</i>	<i>GREEN checkmarks</i>
<i>Security/compliance</i>	<i>"EEEEF8" (right panel)</i>	<i>"D8D8EC"</i>	<i>BLUE</i>

Note: circle fills may use the accent colors above, but glyph icons themselves are always #2682FD (Section 4).

12. Content Rules

What goes on slides:

- *Titles (noun phrases, no verbs where possible)*
- *Sub-labels (one sentence max, no em dashes)*
- *Bullet items (fragments, not full sentences)*
- *Stats (numbers with units)*
- *Section labels (all caps, bold, no letter-spacing)*

What never goes on slides:

- *Em dashes: use colons, commas, or periods instead*
- *Sales questions ("Are you comfortable...", "Shall we...")*
- *Coaching/speaker notes ("Listen first", "Only show if asked")*
- *Full sentences in bullet points*
- *Exclamation marks*

Bullet format: " · Item text" (middle dot + two spaces). Never use Unicode bullet •.

13. Complete Helper Functions

Copy these verbatim at the top of any new deck script:

JavaScript

```
const sh = () => ({ type:"outer", color:"000000", blur:8,  
offset:2, angle:45, opacity:0.06 });
```

```
const sh2 = () => ({ type:"outer", color:"000000", blur:14,  
offset:3, angle:45, opacity:0.09 });
```

```
const circ = (s, x, y, r, fill) =>
```

```
  s.addShape(pres.shapes.OVAL, { x, y, w:r, h:r,  
fill:{color:fill}, line:{color:fill} });
```

```
const circT = (s, x, y, r, t, sz) =>
```

```
  s.addText(t, { x, y, w:r, h:r, fontSize:sz||12, fontFace:"Google  
Sans Flex",
```

```
    bold:true, color:"FFFFFF", align:"center", valign:"middle",  
margin:0 });
```

```
const hline = (s, x, y, w, col) =>
```

```
  s.addShape(pres.shapes.LINE, { x, y, w, h:0,  
line:{color:col||"D8D8EC", width:0.5} });
```

```
const varrow = (s, x, y1, y2, col) =>
```

```
  s.addShape(pres.shapes.LINE, { x, y:y1, w:0, h:y2-y1,  
    line:{color:col||"2682FD", width:1.5, endArrowType:"arrow"}  
});
```

```
const harrow = (s, x1, x2, y, col) =>
```

```
  s.addShape(pres.shapes.LINE, { x:x1, y, w:x2-x1, h:0,
```

```

    line:{color:col||"9090B0", width:1, endArrowType:"arrow"} });

const plusIcon = (s, x, y) => {

  s.addShape(pres.shapes.ROUNDED_RECTANGLE, { x, y, w:0.28,
h:0.28,

    fill:{color:"2682FD"}, line:{color:"2682FD"}, rectRadius:0.05
  });

  s.addText("+", { x, y, w:0.28, h:0.28, fontSize:13,
fontFace:"Google Sans Flex",

    bold:true, color:"FFFFFF", align:"center", valign:"middle",
margin:0 });

  };

const card = (s, x, y, w, h, bg, bd, rad, bdw) =>

  s.addShape(pres.shapes.ROUNDED_RECTANGLE, { x, y, w, h,

    fill:{color:bg||"F4F4F8"}, line:{color:bd||"E4E4F0",
width:bdw||0.5},

    rectRadius:rad||0.18, shadow:sh() });

const sectionLabel = (s, label) =>

  s.addText(label, { x:0.45, y:0.28, w:9.1, h:0.26, fontSize:9.5,

    fontFace:"Google Sans Flex", bold:true, color:"2682FD",
margin:0 });

const slideTitle = (s, title, sub, hasLabel=true) => {

  const ty = hasLabel ? 0.56 : 0.32;

```

```

s.addText(title, { x:0.45, y:ty, w:9.1, h:0.52, fontSize:30,
  fontFace:"Google Sans Flex", color:"12123A", margin:0 });

if (sub) s.addText(sub, { x:0.45, y:ty+0.56, w:9.1, h:0.32,
  fontSize:12.5,
  fontFace:"Google Sans Flex", color:"4A4A6A", margin:0 });
};

// Image helper: fits an image inside a box, centered, preserving
// aspect ratio.

// Requires a dims map of { name: [width_px, height_px] }
// generated at asset-prep time.

const fitImg = (s, name, bx, by, bw, bh) => {
  const [iw, ih] = dims[name];
  const r = Math.min(bw / iw, bh / ih);
  const w = iw * r, h = ih * r;
  s.addImage({ path: assetPath(name), x: bx + (bw - w) / 2, y: by
+ (bh - h) / 2, w, h });
};

```

Also required at presentation setup:

```

JavaScript
pres.theme = { headFontFace: "Google Sans Flex", bodyFontFace:
  "Google Sans Flex" };

```

14. Overflow Prevention Checklist

Before finalising any slide, verify:

- Card bottom: $\text{card_y} + \text{card_h} \leq 5.32''$
 - Last item in card: $\text{last_item_y} + \text{last_item_h} \leq \text{card_y} + \text{card_h} - 0.08''$
 - Title wrapping: at 30pt, up to ~43 characters fit on one line across the full 9.1" width. Longer titles: reduce `fontSize` (26pt handles ~50 characters) rather than splitting into two text boxes.
 - Card title wrapping: at 13pt in a 2.12" wide card, strings over ~20 chars may wrap. Rule: if card width $\leq 2.5''$, card title `fontSize` must be $\leq 13\text{pt}$, and give the title box enough height for two lines before placing the next element.
 - Stat numbers: "Hundreds" at 16pt in a 1.6" wide column is the minimum safe size. Never use more than 20pt for a stat label in a column narrower than 1.5".
 - Subtitle clearance: cards must start at $y \geq 1.56''$ when a subtitle is present, $y \geq 1.22''$ with title only.
 - Arrow overlap: horizontal arrow connectors between cards must have $x1 = \text{card_right} + 0.02''$ and $x2 = \text{next_card_left} - 0.02''$. Never let arrow shapes touch card boundaries.
 - Text next to pills or stats: the text box must end $\geq 0.15''$ before the pill/stat column starts. Never let a description box extend underneath a pill.
 - Pills never wrap: widen the pill or shorten the label.
 - Icon visibility: confirm no icon asset is white-on-transparent before placing it on a light card (see Section 4).
 - Icon/logo parity: in any flow diagram or card group, all boxes have an icon/logo or none do, at identical position and size (see Section 10.6).
 - Cover and divider slides match the fixed format in Section 9.6 (collage art, bold title, no cards).
-

15. Naming and File Conventions

- Output file: `/mnt/user-data/outputs/Axya_[DeckName].pptx`
- Build script: `/home/claude/axya_[deckname].js`
- After build, always: `python /mnt/skills/public/pptx/scripts/rezip.py [file]`
- Then run the theme XML font scrub (Section 3) and rezip again
- Preview: `python /mnt/skills/public/pptx/scripts/office/soffice.py --headless --convert-to pdf [file] then pdftoppm -jpeg -r 120 [pdf] /home/claude/slide_prefix`
- LibreOffice substitutes Arial for Google Sans Flex in previews; this does not reflect actual PowerPoint rendering

