

ZIKUU Mini â?? Admin Dashboardã??Swiftã•§å?•å®?è£ ã•?ã??

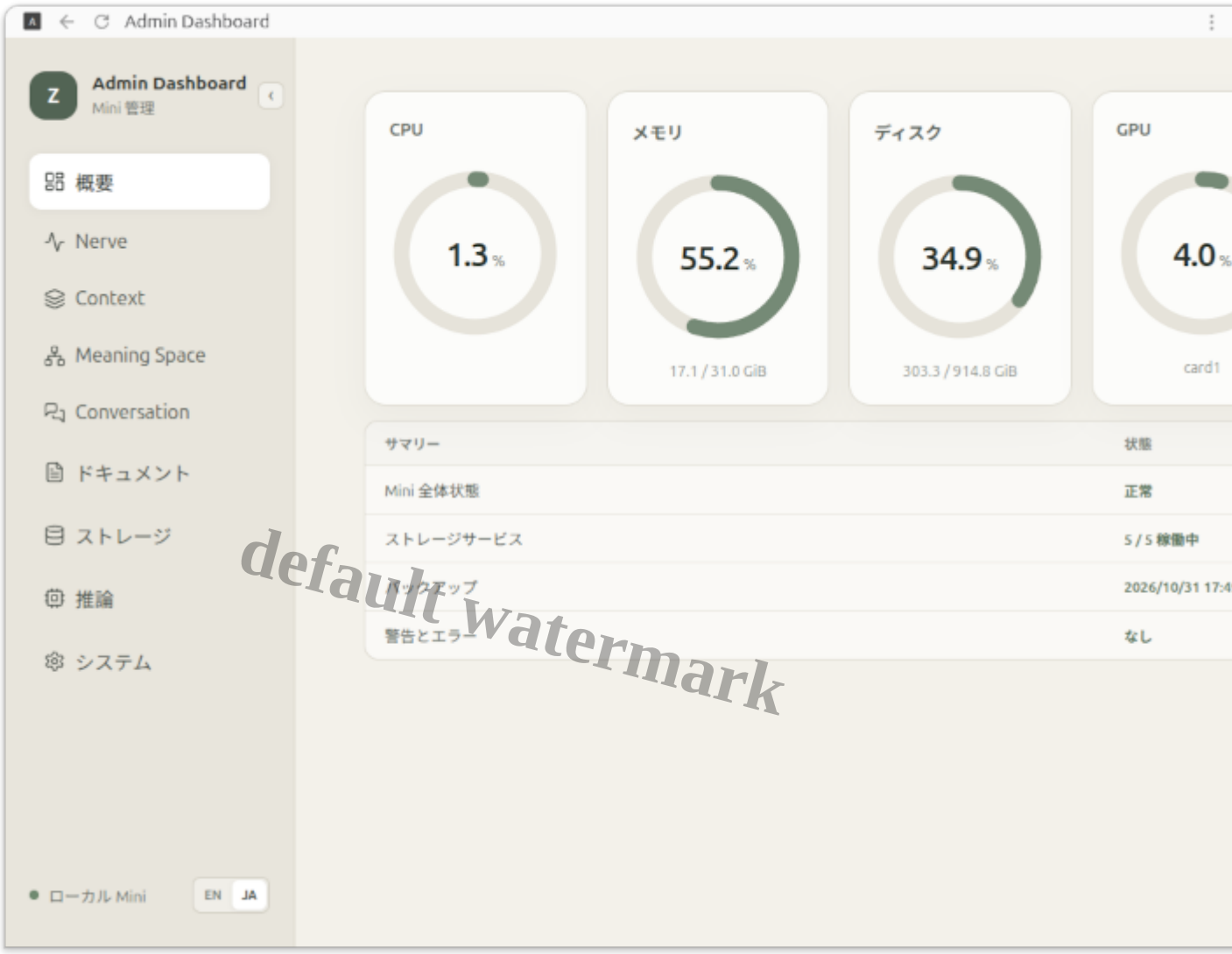
Description

default watermark

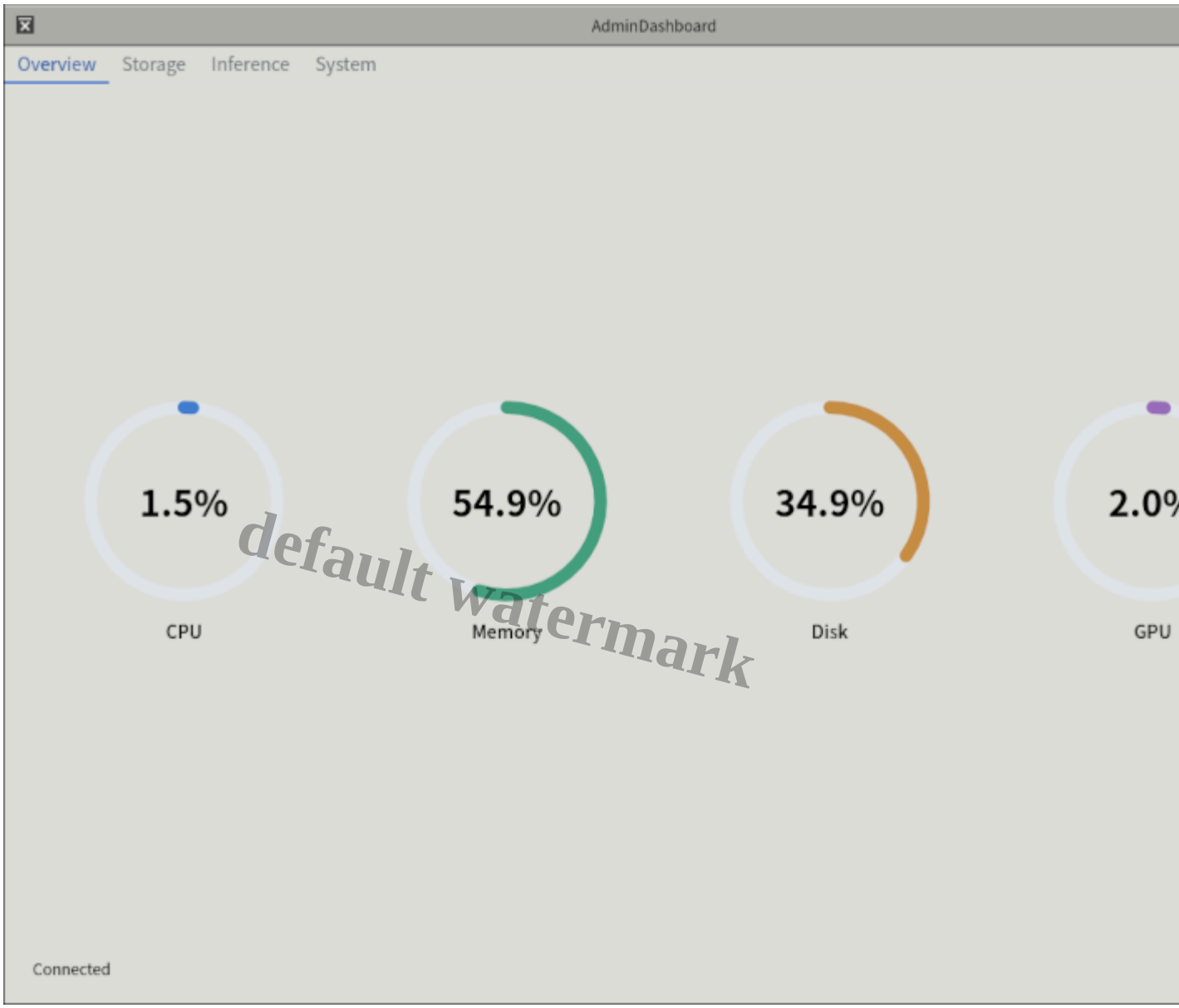
Swiftç?? AppKitç?¸å½?ã•®ã??ã?¬ã?¼ã?¨ã?¯ã?¼ã?¯ã?•ã••ã•?ã•®ã•§ã?•ã•?ã??ã•¾ã•§Reactã??ã?-ã?³ã??ã?¨ã?³ã?? / FastAPIã?•ã??ã?¯ã?¨ã?³ã??ã¨ã•?ã•¦å®?è£ ã•?ã•¦ã•?ã•? ZIKUU Miniç?¨ã•®Admin Dashboardã?? Swift ã•§æ?¸ã••æ•?ã•?ã??ã•?ã•¨ã•«ã•?ã•¾ã•?ã•?ã??

ã•¾ã•?ã•¯ã?·ã?¹ã??ã? æ? å ±ã??å?å¾?ã•?ã•¦è¡¨ç¤ºã•?ã??ç?»é•¢ã•?ã??ç•?æ??ã??

æ¯?è¼?ã®ã•?ã?•ã•« React / FastAPIç??ã•®ç?»é•¢ã•§ã•?ã??



Swiftã•§å?•å®?è£ ã•?ã•?ã??ã•®ã•?ã•?ã??ã??



MetricGaugeViewã•¨ã•?ã•?Viewã•®ã?µã??ã?¯ã?©ã?¹ã??ä½?ã£ã•¦å?•å?©ç?¨å•¯è?½ã«ã•?ã•¾ã•?ã•?ã??

ã?³ã?¼ã??ã•¯ã?ã??ã•ªæ??ã•?ã??

```
 // ~/workspace/AdminDashboard/Sources/AdminDashboard/MetricGaugeView.swift

import Application
import Core

/// A circular utilization gauge for percentage metrics.
final class MetricGaugeView: View {

    var value: Double? {
        didSet {
            setNeedsDisplay()
```

```
    }
}

var trackColor = GraphicsColor(
    red: 0.87,
    green: 0.89,
    blue: 0.91
) {
    didSet {
        setNeedsDisplay()
    }
}

var progressColor = GraphicsColor(
    red: 0.25,
    green: 0.49,
    blue: 0.81
) {
    didSet {
        setNeedsDisplay()
    }
}

var lineWidth: Float = 10 {
    didSet {
        setNeedsDisplay()
    }
}

override init(frame: Rect) {
    value = nil
    super.init(frame: frame)
}

override init() {
    value = nil
    super.init()
}

override func draw(
    _ dirtyRect: Rect,
    in context: GraphicsContext
) {
    let bounds = self.bounds

    let diameter = min(
        bounds.size.width,
        bounds.size.height
    )

    guard diameter > 0 else {
        return
    }

    let strokeWidth = min(
```

default watermark

```
    max(1, lineWidth),
    diameter / 4
)

let radius = (diameter - strokeWidth) / 2 - 2

guard radius > 0 else {
    return
}

let center = Point(
    x: bounds.midX,
    y: bounds.midY
)

context.saveGraphicsState()
defer {
    context.restoreGraphicsState()
}

context.setLineWidth(strokeWidth)
context.setLineCapStyle(.round)

// MARK: Background Track

let track = BezierPath()

track.appendArc(
    center: center,
    radius: radius,
    startAngle: -90,
    endAngle: 270
)

context.setStrokeColor(trackColor)
context.stroke(track)


guard let value, value.isFinite else {
    return
}

let percentage = min(
    100,
    max(0, value)
)

guard percentage > 0 else {
    return
}

let startAngle: Float = -90

let endAngle = startAngle
    + Float(percentage / 100) * 360
```

default watermark

```
        let progress = BezierPath()

        progress.appendArc(
            center: center,
            radius: radius,
            startAngle: startAngle,
            endAngle: endAngle
        )

        context.setStrokeColor(progressColor)
        context.stroke(progress)
    }
}
```

Swiftã•§æ?¸ã••ç?´ã•?ã•?ã?•ã??ã?¯ã?¨ã?³ã??ã•®APIã?µã?¼ã?•ã?¼ã•«ã?•ã??ã?¹ã?¯ã??ã??ã??ã?¢ã??ã?ªã•?ã??ã?¢ã?¯ã?»ã?¹æ§?æ?•ã•«ã•?ã•¦ã•?ã•¾ã•?ã??

å®?æ??ç??ã•«ã?µã?¼ã?•ã?¼ã•«ã?•ã?¼ã?ªã?³ã?°ã•?ã??é?¢ä¿?ã•§ã?•å?¥ã•®ã?¹ã?¬ã??ã??ã•§APIã?µã?¼ã?•ã?¼ã•«ã?¢ã?¯ã?»ã?¹ã?ã?ã??ã•?ã•«ã•?ã•¾ã•?ã•?ã??ã••ã•®ã•?ã?•ã?•ã??ã?¬ã?¼ã? ã?¯ã?¼ã?¯ã®Applicationã•®runLoopã•®ä¸ã•§ã?•ã?¹ã?¬ã??ã??å?¦ç•?ã??ã•?ã??ã??ã•?ã•«æ?¡å¼µã•?ã•¾ã•?ã•?ã??
æ?©é??ã?•ã??ã?¬ã?¼ã? ã?¯ã?¼ã?¯å´ã??ç·´ã??ã•?ã•¨ã•?ã•§ã••ã•¾ã•?ã•?ã??

default watermark

Category

1. æ?¥ã? ã•®æ´»å??

Tags

1. Linux
2. OPENSTEP
3. Swift
4. ZIKUU Mini
5. ä¸?é??å??å¸?
6. å±±æ¢¨ç??

Date Created
2026å¹´11æ??4æ?¥
Author
kazuo-tsubaki