```
- (PMKPromise *)promise {
    PMKAlertViewDelegater *d = [PMKAlertViewDelegater new];
    PMKRetain(d);
    self.delegate = d;
    [self show];
    return [PMKPromise new:^(id fulfiller, id rejecter){
        d->fulfiller = fulfiller;
    }];
}
```

```
- (PMKPromise *(^)(id))then {
    return ^(id block){
        return self.thenOn(dispatch_get_main_queue(), block);
    };
}
```

```
- (PMKResolveOnQueueBlock)thenOn {
    return [self resolved:^(id result) {
        if (IsPromise(result))
            return ((PMKPromise *)result).thenOn;

        if (IsError(result)) return ^(dispatch_queue_t q, id block) {
            return [PMKPromise promiseWithValue:result];
        };

        return ^(dispatch_queue_t q, id block) {

            // HACK we seem to expose some bug in ARC where this block can
            // be an NSStackBlock which then gets deallocated by the time
            // we get around to using it. So we force it to be malloc'd.
            block = [block copy];

            return dispatch_promise_on(q, ^{
                return pmk_safely_call_block(block, result);
            });
        };
    }
    pending:^(id result, PMKPromise *next, dispatch_queue_t q, id block, void (^resolve)(id)) {
        if (IsError(result))
            PMKResolve(next, result);
        else dispatch_async(q, ^{
            resolve(pmk_safely_call_block(block, result));
        });
    }];
}
```

```
[self resolved:^PMKResolveOnQueueBlock(id result) {

} pending:^(id result, PMKPromise *next, dispatch_queue_t q, id block, void (^resolver)(id)) {

}];
```